TU WIEN Informatics

Manual Analysis and Early Identification of At-Risk Students in Programming Courses

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Jovan Dragojlovic, BSc
Matrikelnummer 11910030

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr. Thomas Grechenig

Wien, 8. Mai 2024

Unterschrift Verfasser

Unterschrift Betreuung

Manual Analysis and Early Identification of At-Risk Students in Programming Courses

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Jovan Dragojlovic, BSc
Registration Number 11910030

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr. Thomas Grechenig

Vienna, 8th May, 2024

Signature Author

Signature Advisor

Manual Analysis and Early Identification of At-Risk Students in Programming Courses

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Jovan Dragojlovic, BSc
Matrikelnummer 11910030

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr. Thomas Grechenig

Wien, 8. Mai 2024

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Jovan Dragojlovic, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 8. Mai 2024

Jovan Dragojlovic

Danksagung

Mit Freude möchte ich mich bei meiner Familie und Freunden für die Unterstützung während meines Studiums auf der Technischen Universität Wien bedanken. Mein Dank gilt insbesondere meinen Freunden, die diese Masterarbeit korrekturgelesen haben und mir essentielle Ratschläge und Vorschläge zu meiner Arbeit gegeben haben.

Außerdem bedanke ich mich herzlich bei dem technischen Support Team der Technischen Universität Wien und Dr. Rakoczi Gergely, die mir erweiterte Berechtigungen zum TUWEL-Kurs erteilt haben und mich bei Fragen rund um die genannte Online-Plattform unterstützt haben.

Zudem möchte ich mich vor allem auch bei Dr. Stefan Podlipnig für die kompetente Unterstützung bedanken, der mich nicht nur beraten hat die Case Study über den Kurs Einführung in Programmierung 1 zu erstellen, sondern auch Daten und Informationen zu vorherigen Semestern zur Verfügung gestellt hat.

Zu guter Letzt möchte ich mich für die gewissenhafte Unterstützung meines Betreuers Thomas Grechenig und seinen Projektassistenten bedanken, die meine Arbeit zu dem ausgewählten Thema zur Verfügung gestellt haben und mir zusätzlich bedeutende Anregungen gaben.

Acknowledgements

I would like to thank my family and friends for their support during my studies at the Technische Universität Wien (TU Wien). Especially, I would like to thank my friends who proofread this master thesis and gave me essential advice and suggestions on my work.

I would also like to thank the technical support team of the TU Wien and Dr. Rakoczi Gergely, who gave me extended permissions to the TUWEL course and supported me with questions about the online platform mentioned previously.

Additionally, I would also like to thank Dr. Stefan Podlipnig for his competent support, who not only advised me on how to create the case study for the Introduction to Programming 1 course, but also provided me with data and information on previous semesters.

Last but not least, I would like to thank my supervisor Thomas Grechenig and his project assistants for their conscientious support, who made my work on the selected topic available and gave me additional important suggestions.

Kurzfassung

Die Bedeutung und Nachfrage von Experten in der Informationstechnologie (IT) hat in den letzten Jahren am modernen Arbeitsmarkt an Bedeutung gewonnen. Um nun das Angebot von kompetentem IT-Personal zu gewährleisten, bieten höhere Schulen und Universitäten, wie die Technische Universität Wien, einführende Programmierkurse an. Aus der Literatur [1] geht hervor, dass verglichen mit anderen Kursen die Durchfallquote höher ist. Gründe dafür sind vielfältig, besonders Lehrveranstaltungen mit vielen Teilnehmer*innen sind ohne elektronische Hilfsmittel kaum überschaubar.

Mit diesem Grundproblem beschäftigt sich diese Masterarbeit hauptsächlich, welches in den Forschungsbereich „Educational Data Mining“ fällt. In erster Linie verwenden viele Bildungsstätten, unter anderem die Technische Universität Wien, bereits wohletablierte Lern-Management-Systeme wie Moodle, die eine Datenverwaltung von Studierenden vereinfachen sollen. Moodle bietet eine Vielzahl von Speicher-, Einstellungs- und Analysemöglichkeiten an. Durch die standardisierte Gestaltung dieser digitalen Plattform, gibt es jedoch wenig Konfigurationsspielraum für erweiterte Analysen.

Das Ziel dieser Masterarbeit ist es nun mit einem Prototyp eine erweiterte visuelle, sowie statistische Analyse in einem universitären Umfeld zu ermöglichen, welche die Basis für neue Erkenntnisse bietet, um somit die Ressourceneinteilung der Bildungsstätten zu optimieren. Vom erhöhten Überblick profitieren nicht nur Mitarbeitende einer Universität, sondern auch Studierende, durch beispielsweise intensivere Betreuung.

Basierend auf Anforderungen von Leiter*innen einer Lehrveranstaltung wurde dieser Prototyp entwickelt – ein Vergleich der derzeitigen Einschätzung und einer quantitativen Analyse zeigte ebenso die Wirksamkeit des Prototyps. Mithilfe einer Fallstudie wurden die Möglichkeiten, sowie Grenzen des Prototyps ermittelt.

Schließlich wurde das System sehr dynamisch definiert und ausführlich dokumentiert, um eine Weiterentwicklung, Reproduktion und Anpassung zu ermöglichen. Es optimiert den Prozess der Entscheidungsfindung mit zusätzlichen Informationen, um letztendlich Studierende in ihrem Programmierwissen effektiver auszubilden und ressourcensparend zu agieren.

Keywords: *Educational Data Mining, Datenvisualisierung, Korrelationsanalyse, Datengewinnung, API, Informatikausbildung, Computer Science 1*

Abstract

The importance of and demand for experts in information technology (IT) has increased in the modern labor market in recent years. In order to ensure the supply of competent IT personnel, secondary schools and universities, such as the TU Wien, offer introductory programming courses. The literature [1] shows that the failure rate is higher compared to other courses. There are several reasons for this, especially courses with many participants are hardly manageable without electronic aids.

This master's thesis mainly deals with this basic problem, which falls within the research area of „Educational Data Mining". First and foremost, many educational institutions, including the TU Wien, already use well-established learning management systems such as Moodle, which are designed to simplify student data management. Moodle offers a wide range of storage, setting and analysis options. However, due to the standardized design of this digital platform, there is little room for configuration for advanced analyses.

The aim of this master's thesis is to create a prototype to enable an extended visual and statistical analysis in a university environment, which provides the basis for new findings in order to optimize the allocation of resources at educational institutions. Not only university employees benefit from the increased overview, but also students, for example through more intensive support.

This prototype was developed based on requirements of course-leaders - a comparison of the current assessment and a quantitative analysis also showed the effectiveness of this prototype. Supplementary, a case study was used to determine the possibilities and limitations of the prototype.

Finally, the system was defined very dynamically and documented in detail to enable further development, reproduction, and adaptation. It optimizes the decision-making process with additional information in order to ultimately train students more effectively in their programming knowledge and save resources accordingly.

Keywords: *Educational Data Mining, Data Visualization, Correlation Analysis, Data-Mining, API, Computer Science Education, Computer Science 1*

Contents

CHAPTER 1

Introduction

Several research departments such as Statista [2] argue that „leading companies across industries push for further adoption of digital technologies, IT professionals have become some of the most in-demand members of today's labor force.“ Additionally, the rising number of artificial intelligence, robots, computer systems or simple technological devices further increases the demand for IT experts [3]. Thus, universities and other educational facilities provide introductory programming courses, which are also called CS1 in recent literature [4]–[6]. Even the scientific field is influenced by the technological development by using data mining techniques or principles to improve the overall quality of education.

Therefore, the research area is called Educational Data Mining (EDM), which aims to support lecturers, tutors, instructors, or institutions in decision-making processes [7]. One of the major goals of this research field is to accurately predict students' performance and possible difficulties or obstacles within their studies. For instance, creating or developing models to predict or indicate the students' performance by using a variety of approaches.

Definition 1 ***Educational Data Mining*** *(EDM) is an emerging discipline, which explores large-scala data mainly from educational sources. Within this approach student data will be used to fetch valuable information about the students for example in an introductory course or training [8].*

In recent years, the interest in EDM-approaches is quickly increasing due to possible positive outcomes and benefits such as overall improvement of the teaching quality. This aspect is not only beneficial for students, but also for all other stakeholders like IT companies, other research areas, and even instructors or lecturers [9], [10]. Moreover, the additional information helps to organize internal resources, such as staff, books, or licenses more efficiently. In most cases, EDM [11] focuses on the student's academic performance, for example by either calculating the probability of dropping out, failing the course, or predicting the final grade of each student.

According to M. M. Jamjoom et al. [7] the rising demand of IT or programming services also increases the amount and level of specialization. However, the authors argue that these applicants are lacking of important skills, knowledge, and qualifications to further proceed in these areas. Improving the quality and support during the lecture or training could be a good way to tackle the roots of the problem and therefore teach the needed capabilities to students. In the following sections, the overall topic of the master thesis will be described in more detail. Section 1.1 provides information about the underlying problem, which might be solved by the end of this work. The next Section 1.2 and 1.3 are about the personal motivation and why the mentioned problem statement is relevant and further describes the expected results respectively. Finally, section 1.4 and 1.5 represents a short overview of the thesis' structure.

1.1 Problem Statement

Comparably, programming courses appear to be on the low end of pass rates. According to the report of the working group ITiCSE [1] pass rates in introductory programming courses average about 67% to 75%. Thus, more than 1 out of 4 programming students fail introductory courses - students who fail either retake the whole course, switch their major or drop out completely. Similar findings could be observed at the University of Technology in Vienna - every semester about 20% to 30% of students fail the *Introduction to Programming 1* course.

C. Gordon et al. [4] argue that computer science instructors or lecturers inquired such an indication or prediction system to already intervene in an early stage or adjust the lecture's content accordingly. In other words, there is a lack of insights on reports and statistics to allocate internal resources and staff accordingly. Developing a model to provide such information would not only decrease the students at risk of dropping out but also improve the overall quality of the course. Nevertheless, this problem poses as a major challenge, especially in large courses with hundreds of participants - it is hardly possible to keep track of every student's concerns or issues.

Thus, this thesis seeks for a way to assist instructors, lecturers, and other university staff by for example providing visual warnings or indicate common patterns. This indication or prediction model could also be beneficial for online courses, online trainings and during pandemics, in which students are not able to get in touch with course staff or peers as easily [12], [13]. To narrow down the focus of the master thesis the following research questions were created in close cooperation with Dr. S. Podlipnig, who is mainly responsible for the *Introduction to Programming 1*[1] course at the University of Technology in Vienna [14].

RQ1 : „Research Question 1: To what extend can students at risk of failing a course be determined by previous activities?“

RQ2 : „Research Question 2: Which combination of attributes is most relevant to approximate a student's final score?“

RQ3 : „Research Question 3: Which attributes or properties of course-contents or taught programming skills are hard to comprehend for computer science students?“

The respective master thesis tries to tackle these problems by creating a model to indicate or visualize which students might need more attention in programming courses.

Another issue of this thesis will be the replicability and long-term support of the provided tools, which was a major concern for Dr. S. Podlipnig — creating a generic, well-documented and easy to use tool is also essential for future work and the EDM-community. Additionally, due to the changing requirements, the tool must be configurable and extensible — these properties ensure a long-term use and further development of, for example, new features.

1.2 Motivation

Computer programming or developing computer science skills are increasingly demanded in various workplaces and applications. Which means effective teaching of these skills will be a prerequisite for many fields of study such as *Introduction to Programming 1*, *Software-Engineering and Project Management* and *Introduction to Security* at the University of Technology in Vienna. For tutors, instructors, and lecturers it became noticeable that some students have retaken the mentioned courses, which additionally increased the amount of work per staff-member for the next course.

Especially, in large courses with several hundred participants such as *Introduction to Programming 1* and *Software-Engineering and Project Management* it is quite hard to keep track of all students and their issues or concerns. For instance, in the winter term of 2022 eight course-leaders and 29 tutors tried to answer open questions, help with programming issues, and also lecture about theoretical basics in computer science. Nevertheless, only about 77% of students actually passed the *Introduction to Programming 1*.

[1] `https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=6587&dsrid=5&courseNr=185A91&semester=2023S` [last access: 25.12.2023]

Therefore, for tutors and lecturers it was quite hard to estimate the performance of students by using the available tools like Technische Universität Wien E-Learning Platform (TUWEL)'s reporting services and visualizations. As mentioned, in close cooperation with Dr. S. Podlipnig the master thesis tries to find a lightweight model to track student's performance during the ongoing semester to determine which student's will be at risk of dropping out. The fetched data should be based on existing data and no additional tracking or analytics tools should be used for this purpose to keep the model lightweight and easy to setup and maintain.

Definition 2 ***TUWEL*** *stands for the „****T****echnische* ***U****niversität* ***W****ien* ***e-L****earning Platform"*[2]*, which is a Moodle-instance [15] extended and modified by technicians in the University of Technology in Vienna for specific purposes to adapt and contribute to an improved user-experience for students, teachers and other users.*
The TUWEL-team is continuously developing and fixing existing issues and keeps track of network traffic to ensure stable and secure connections to resources, assignments, and additional information.

On top of that, external events and conditions like the COVID-19 pandemic, shortage of university staff and internal resources, further increased the need for such a more efficient approach to indicate students at risk. Nevertheless, these issues also pertain other large courses at the University of Technology in Vienna. Actually, professors of various faculties would also be interested in the development of such an indication model to help to allocate internal resources and staff more efficiently. The initial data for the quantitative analysis can be extracted from the university's management platform TUWEL, which is as mentioned, a Moodle [15] instance.

1.3 Expected Results

Computer programming or software development is a fundamental course in every IT, computer science or business informatics curriculum. Furthermore, more and more majors such as mathematics, engineering, physics, or biology are requiring a basic level of programming skills. Ultimately, a model to improve the quality of programming lectures and distribute workload among lecturers and tutors more efficiently could be beneficial for all stakeholders.

The result of this master thesis will be a software artifact to identify and classify programming students according to their course-activities, possibly also in the following three categories:

[2] https://www.tuwien.at/tu-wien/organisation/zentrale-bereiche/campus-software-development/lehr-und-lerntechnologien/services/tuwel [last access: 27.12.2023]

- students, who are at risk of dropping out or struggle to achieve the required points to pass the programming course.
- students, who are (theoretically) able to reach the minimum required points to pass the programming course.
- students, who already have enough points to pass the programming course.

However, these findings and mentioned categories presuppose certain rules and grading scheme, which will be explained in Section 2.2. The visualization of the fetched data can further be analyzed and interpreted by lecturers, instructors, or tutors. Thus, this model enacts more like an assistance-system for lecturers, providing valuable information and visualizations based on a generic design and configuration. The primary application would be for large courses with many participating students and a rather complex structure - favorably most of the information is stored on one site to ensure a lightweight use of the system.

On top of that, the developed tool should be able to identify difficult course-content or specific sections, which might cause problems for students or might be hard to comprehend. Thus, countermeasures such as revisions or extra classes can be set up for computer science students to additionally teach these course contents and answer open questions. This part could contribute to less dropouts or generally less students at risk of dropping out, which conversely decreases the amount of work for tutors, lecturers, and trainers. This aspect might be very important in programming courses, because according to Dr. S. Podlipnig most of the advanced course content is based on previous classes and gained knowledge in these classes of the Introduction to Programming 1 (IP1) course.

Even further inefficiencies in the current grading scheme can be discovered by using the developed model. All in all, this master thesis tries to tackle the previously mentioned problems or issues and increase the quality of the overall lecture and at the same time increase the learnings and knowledge of participating programming students.

1.4 Scientific Contributions

This master thesis aims to develop a data-driven approach in the field of EDM to monitor student's performance and get further valuable insights for lecturers and tutors. Additionally, the visualization tool and model is based on the well-established Management Platform Moodle, which is used by many universities, companies or groups in Europe [15]. In short, due to the generic development, extensive documentation, and focus on usability, the approach could be applied to various courses.

However, this master thesis aims to improve the quality of computer science courses and decrease dropout rates of programming students. The reasons for specifically focusing on programming students are due to the constructive content and steps required to learn a programming language. Furthermore, it might be possible to gain insights about code quality, common syntax errors and programming-related content. After some adaptations, the model could also be used for other non-programming courses, which will not be part of this master thesis.

Ultimately, the hidden information and insights can be categorized and compared respectively, which is not only beneficial for the students, lecturers, or other parties, but also for the EDM-community — the thesis inspects the lack of tool-support and inefficiencies using an exemplary Learning Management System (LMS). This also aligns with the ultimate goal of EDM to explore and further understand different types of data originating from student's behavior or performance. In theory, the gained information could be used to improve the overall lecture or support lecturers, instructors, or tutors during teaching or in subsequent analysis.

1.5 Structure

This section explains the overall structure of this master thesis in more detail by following the approach by F. Cady „Data Science Road Map" [16]. The following Figure 1.1 illustrates the needed steps to firstly deploy the needed code and secondly present the results visually, which includes a manual statistical analysis and interpretation of the outcomes.

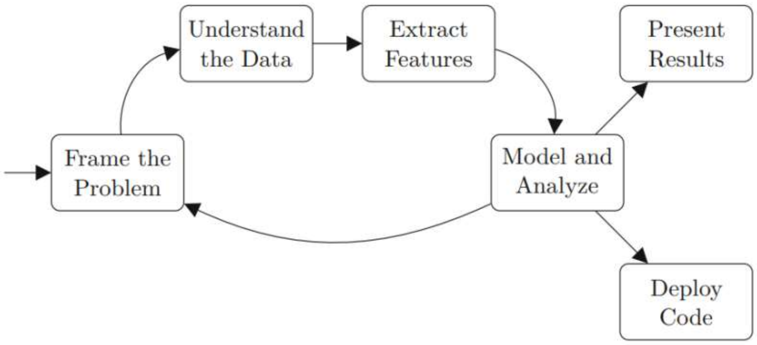


Figure 1.1: Data Science Road Map [16]

Therefore, the arrangement of chapters of this master thesis will be conducted as follows:

- *Chapter 1* is representing the current chapter. It examines the overall problem or issue, which could be solved with this master thesis. Thus, this section also includes the initial motivation, research questions and expected outcomes. Furthermore, the scientific contribution will shortly be explained in this section, which includes the added value of this thesis for the research community of EDM [8]. Lastly, the current sub-section gives insights into the structure of this thesis.

- *Chapter 2* describes all fundamental theories, concepts and terms required for the design science approach of this master thesis. Mainly including domain-specific terms and explain pivotal aspects of the used Management Platform. Further concepts about indicating students at risk, classifications and visualizations will be quoted in this section.

- *Chapter 3* explains the current state of the research, existing tools, and practices. Relevant publications or papers with similar approaches will be primarily stated to avoid mistakes of previous tools and improve the development of the visualization tool and indication model. This chapter will be sub-divided in smaller sections according to the specified research areas such as EDM and data visualization of data.

- *Chapter 4* represents the intended methods and approaches for this master thesis. Thus, dividing the scientific methodology per research question and clearly displaying the connection between the underlying problem and the planned solution-approach. Additionally, the planned evaluation and assessment method will be included in this section to facilitate a solid foundation and a complete overview of the used approach.

- *Chapter 5* stands for one of the core parts of this master thesis by describing the development process and model design in more detail. This view includes all processes, starting from the data extraction and transformation of data to the final visualization. Ultimately, the step-by-step approach and documentation of this chapter intends to create a complete overview of all used tools and techniques to enable replication for future use cases.

- *Chapter 6* focuses on the visual analysis concepts, which are used within this master thesis. Especially, extended information about the case study of the IP1 course and the tool's interactive environment will be provided in this chapter.

- *Chapter 7* summarizes the results and insights of the quantitative analysis and expert evaluation and compares these findings with existing literature. Generally, the chapter can be used for future analysis of courses by using LMSs — technical findings and evaluation of the development process will be listed accordingly. This part of this master thesis will be useful for student staff to assess the capabilities of the developed model, which includes the limitations of using an LMS as a management tool for university courses.

- *Chapter 8* discusses the findings and new insights during the case study and the overall research process. Additionally, this chapter includes an outlook on future work in the area of EDM. Lastly, including a summary of the master thesis and the found outcomes, which includes the scientific contribution in the mentioned research area.

CHAPTER 2

Foundations

The following chapter of this master thesis is primarily concerned with a basic overview of the foundations and fundamental information, which sets the base for latter chapters.

2.1 Domain Concepts

Within this master thesis specific tools, approaches, and methods are used, which are specifically pre-defined by the TU Wien or by course-leaders of the respective course. Therefore, the following sections give very brief insights in those topics and domain-specific concepts to get a basic understanding of processes in more advanced chapters.

2.1.1 Programming environment and language

This section briefly describes the used or recommended technologies used in the IP1 course by students as well as additional management tools for tutors, lecturers, and instructors. Moreover, this section will delve into further concepts or techniques to fetch, format, and visualize student data.

IntelliJ IDEA

Fundamentally, IntelliJ IDEA [17] is an integrated development environment (IDE) mainly for Java and other programming languages such as Python or Kotlin. The software development company JetBrains provides a set of tools for coding, testing, and debugging. The IP1 course and more advanced courses of the informatics curriculums in the TU Wien recommend using IntelliJ IDEA for programming. JetBrains even offers the Ultimate edition for students and lecturers[3], which contains additional features and libraries. This includes further frameworks and plugins that can be installed to increase the overall

[3] https://www.jetbrains.com/community/education

productivity of students, while at the same time being user-friendly, which is crucial for beginners.

Java

The programming language and computing platform Java [18] was initially released in 1995. It is continuously evolving since then and extending its features and capabilities — according to Statista, Java is one of the most used programming languages worldwide as of 2023 [19]. Java [18] comes with many services and already built-in features, which established a suitable platform to learn programming basics like loops, conditions, recursion and arrays. This programming environment also provides advanced integrated libraries and features such as Object-oriented Programming (OOP) and lambda-expressions. Additionally, Java is free for personal use and development and many applications are based on this computing platform.

2.1.2 Distributed Version Control System

This section gives a very brief overview of the basic concepts, which represent the foundation for the tools or platform used in programming courses and the selected case study. In order to track, manage and organize files, grades, and feedback, computer scientists use Version Control Systems (VCSs) or Revision Control Systems (RCSs) [20]–[22], which are also commonly used in software development or coding projects. Some of its features enables easier and more transparent tracking of progresses and histories. Thus, these systems include a collaborative framework to a central repository, while being connected via private or public network. Distributed Version Control Systems (DVCSs) represents a type of a VCS, which exhibits local repositories for every user, who can work completely offline as seen in Figure 2.1.

In other words, each user has a copy of the central repository — changes or modifications must be pushed or pulled over networks. Advantages of DVCSs are that teams can work from different workplaces, countries, time zones, and can create own branches before pushing the changes over a network.

For example, lecturers, tutors, and instructors utilize these benefits to add, delete, modify, or move study-related files. This is especially useful for systems in dynamic environments in which many changes and modifications are made by various contributors. Tools such as GitLab (see Section 2.1.2) utilize these frameworks to central repositories for effectively tracking source code and project progression.

Software Repository

Software repositories [23] are closely linked to VCS and operate as central storage locations to save files, programs, projects, documentation, and any kind of historical data. According to O. M. Khanday and S. Dadvandipour [23] the data can be categorized in the following types:

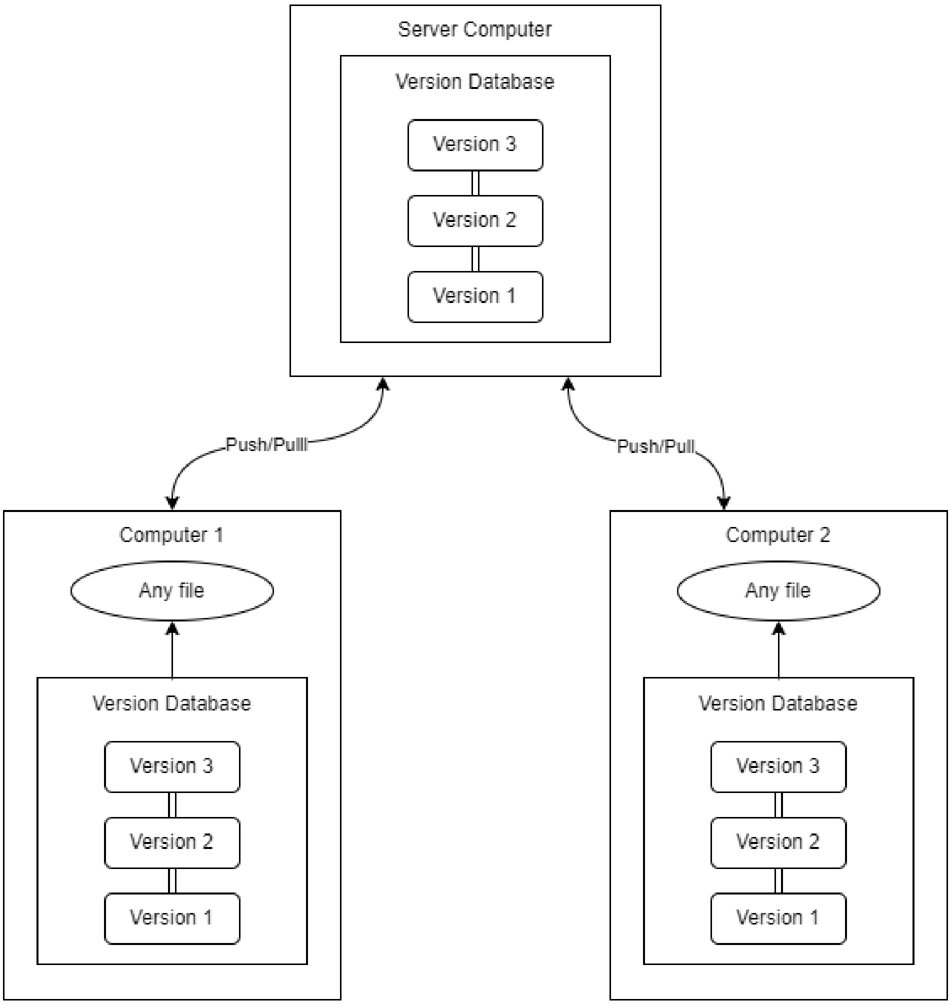


Figure 2.1: Distributed Version Control Systems [22]

1. *Unstructured data*: These files often contain natural language or texts such as reports, descriptions, or documentations.

2. *Structured data*: These files include source code, structured graphs, logs, or any kind of databases.

This master thesis will deal with both types of software repositories, however mostly structured data will be suitable for further analysis. Eventually, data can be pushed into online-repositories, which overwrites the made modifications, such as adjusted source codes or edited files to the central online-repository. In the case of this master thesis, adjustments in student grades can be uploaded or pushed into the central repository accordingly. This repository is only available for students, instructors, and lecturers of the affected course. Also, data can be fetched by pulling the current version from the mentioned online-repository.

GitLab

GitLab [24] is based on a instance of a DVCS called Git [22], which is widely used for tracking differences in source code, files, and the area of software development. GitLab started 2011 as an open-source project to establish a collaboration-platform for programmers by providing free private and open repositories including a wiki, branch trees, issue boards, and many more useful features. Many tasks for the management and planning of projects and additional monitoring and security enables a more effective collaboration and productivity. The version of GitLab, which is used in the lecture is open-source, thus fully free and transparent. The used Enterprise Edition was provided under the MIT License.

2.1.3 Tools used for modeling

Further tools, which are used for the development of the visualization tool will be listed below. Inherently, the workflow was split in three processes:

1. Step: Fetching data by using the provided Application Programming Interface (API) of Moodle's Webservices.

2. Step: Filtering and customizing the data structure for the last step.

3. Step: Visualizing the further filtered data with a well-established and end-user friendly tool.

Moodle Webservices API

The management platform Moodle also provides a Web Service framework [25], which includes valuable features to create various web services for external use. Combined with the External API, Moodle furnishes classes and defines Endpoints to enable use cases for different modules.

The authentication and latter function calls against the relevant API-classes are handled in a multi-step process between the Client, Login Endpoint, External API, and Plugin API. For the first part, to fetch relevant data a TUWEL documentation was provided by the TU Wien-technical support, which includes all possible API-calls, its input parameters, output parameters, and additional information such as expected error messages. This resource might require a valid TUWEL-account and sufficient permissions and must be activated by the TUWEL-technical support.

Postman

The Postman API Platform[4] was used to build and utilize Moodle's External Services. According to D. Westerveld [26] Postman simplifies the API lifecycle and establishes a platform for online collaboration with co-workers, partners, and other stakeholders. Postman contains an intuitive Graphic User Interface (GUI), which offers a variety of configurations to automate and configure API calls.

Within this master thesis Postman was used to test and validate the TUWEL's API documentation and get first insights in the provided student and course information. Thus, this software program can be also used to filter non-functioning, defective, or non-informative API calls efficiently by using placeholders and a pre-defined Postman Collections[5]. These Collections are widely used to link API elements and enable the reuse of variables for any GET-, POST-, PUT-, OR DELETE-Requests.

Python and Jupyter Notebook

According to Statista [19], Python [27] is currently one of the most used programming languages among software developers. It is known for its simplicity, extensibility, and effective approach to object-oriented programming, which facilitates and improves fast development of scripts and various applications. Additionally, due to many contributors and Python's interpreted nature, a large amount of free-to-use libraries can be used for programming or as extensions to other tools.

One of these more advanced extensions or plugins is the Jupyter Notebook [28], which is originally a web application for sharing documents over networks. Thus, by using this plugin one can combine Python-code with Markdown[6] text, which creates a document-centric experience. Ultimately, Jupyter Notebook increases the readability and visualizes the workflow and author's approach in many given domains. Conveniently, the input code and output can be saved in the „.ipynb“ file, which enables other parties to read the software-program without ever running it themselves.

PowerBI

Fundamentally, according to L. T. Becker Power Business Intelligence (Power BI) [29] is an analytics and visualizations tool developed by Microsoft[7]. Power BI uses a „Power“ add-on called Power Query, which is also integrated in other Microsoft Office products such as Microsoft Excel. It offers a variety of functions and input-data formats, among these Comma Separated Values (CSV) files, various databases, JSON files, Microsoft Access files, etc. This feature allows a convenient integration of third-party data, while

[4] `https://www.postman.com/` [last access: 29.12.2023]

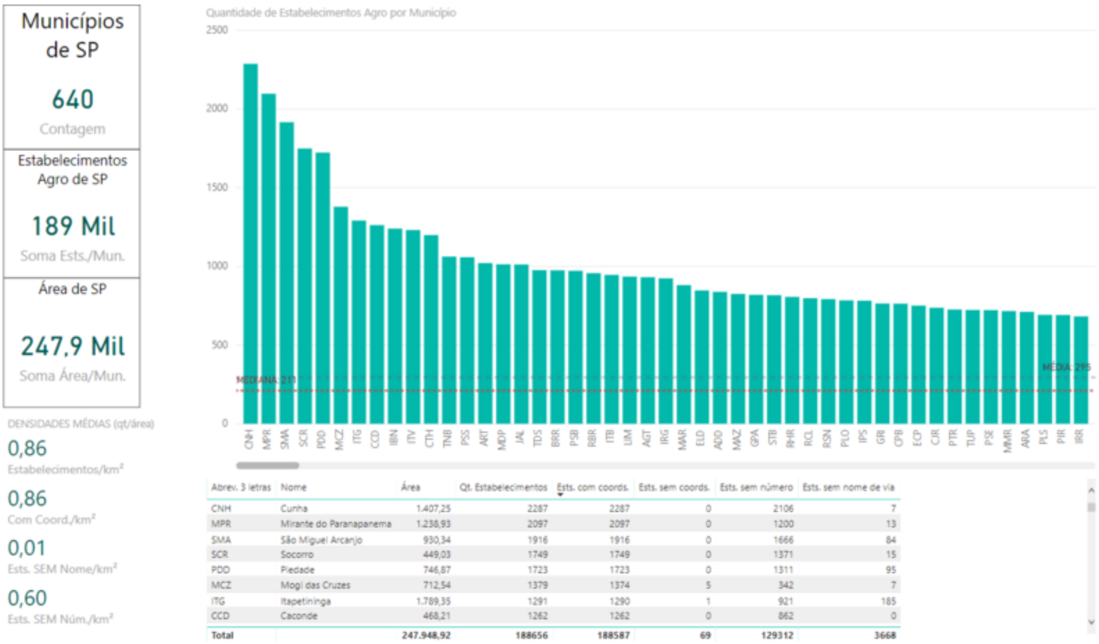
[5] `https://www.postman.com/collection/` [last access: 29.12.2023]

[6] `https://www.markdownguide.org/` [last access: 29.12.2023]

[7] `https://www.microsoft.com/` [last access: 02.01.2024]

at the same time providing support for large-scale databases and additional measures to adapt the data to the requirements.

Thus, tables, columns and, in general data can be reformatted and combined before loading it into the so called „Report View“, which initially offers different types of diagrams. These diagrams can be freely distributed on different pages to outline the important information or data as shown in Figure 2.2.



Abrev. 3 letras	Nome	Área	Qt. Estabelecimentos	Ests. com coords.	Ests. sem coords.	Ests. sem número	Ests. sem nome de via
CNH	Cunha	1.407,25	2287	2287	0	2106	7
MPR	Mirante do Paranapanema	1.238,93	2097	2097	0	1200	13
SMA	São Miguel Arcanjo	930,34	1916	1916	0	1666	84
SCR	Socorro	449,03	1749	1749	0	1371	15
PDD	Piedade	746,87	1723	1723	0	1311	95
MCZ	Mogi das Cruzes	712,54	1379	1374	5	342	7
ITG	Itapetininga	1.789,35	1291	1290	1	921	185
CCD	Caconde	468,21	1262	1262	0	862	0
Total		**247.948,92**	**188656**	**188587**	**69**	**129312**	**3668**

Figure 2.2: Power BI Report View visualization example[8]

Beside the external tools and integrations for other Microsoft applications Power BI also provides a free-to-use standalone Windows application called Power BI Desktop[9], which combines several layers to effectively analyze datasets. Due to its generic and intuitive design, Power BI and its features can be accessed and learnt by using the guide and tutorials provided by Microsoft [30].

2.2 Course Concepts

The following section describes the course IP1[10], which is a part of the mandatory curriculum of every branch of informatics study at the TU Wien. This course will be later

[8] https://commons.wikimedia.org/wiki/File:DadosBI-exemplo3.2-powerBI.png [last access: 02.01.2024]

[9] https://powerbi.microsoft.com/en-us/desktop/ [last access: 02.01.2024]

[10] https://tiss.tuwien.ac.at/course/courseDetails.xhtml?courseNr=185A91&semester=2023W&dswid=8290&dsrid=596 [last access: 04.01.2024]

used to practically examine the summarized concepts and test the developed visualization tool and statistical model.

2.2.1 European Credit Transfer System

The European Credit Transfer System (ECTS) [31] is one of the most commonly used standards for measuring the workload of a student in higher education such as universities. Thus, ECTS describes the planned time to complete a course, which includes contact hours, work- or study-time. According to S. Loskovska a full study year in any European higher education facility is equivalent to 60 ECTS-credits. In practice, one ECTS-credit would be equal to approximately 25-30 hours of student work — to bring this information into context, the workload of the IP1 course was estimated to 5.5 ECTS, which means 137.5-165 hours.

Nevertheless, since the winter term in 2021 the IP1 course became an exception by providing a so-called K3-exam. Only students, who pass the previous K2-exam can participate on the K3-exam subsequently. One is rewarded with 5.5 ECTS-credits directly if the K3-exam is passed, which benefits both, the lecturing-staff and experienced students, who already learnt the content and know how to code. Similarly, an easier K2-exam could be also taken to reduce the number of assignments for students. In general, both exams are optional and only recommended for students with a certain level of programming knowledge, due to requirements for more advanced programming courses at the TU Wien.

2.2.2 Educational Grading Standards

Grading students is one of the core-features in nearly every educational institution - according to J. Schneider and E. Hutt [32] grading systems were introduced to create a measure for comparison between the work of a classroom in relation to the society at a larger scale. In this section of the master thesis the basic concepts and the developments of known grading schemes will be discussed in more detail. Kirschenbaum et al. [33] argue about the psychological effects of grading for children, teenagers, and also adults, which will not be part of this master thesis. However, this master thesis could represent a foundation for future work in this domain.

As much as grading schemes create measures for comparison, it also initiate spaces for constant competition and a meritocracy [32]. In early stages of the European and American republic grades running from four to zero were used to assess for instance Dissertations, Disputes, and Colloquies. Oral examinations based on scores of two earned students passing the mark — generally speaking, no coherent grading schemes were defined in that time. Various universities experimented with grading schemes adapted to the curriculum and course contents. These grades were often kept secret from students, which minimized the amount of transparency, this aspect has improved in most educational systems over the past centuries.

Grades or comparisons were also conducted in relation to other students. In fact, these ranking systems tried finding the highest performers for certain tasks and other students were ranked based on the highest results. According to J. Schneider and E. Hutt [32] constant changes in rankings increased rather the competition among students than increasing the intellectual and moral development. Starting from the late 18th-century authorities such as the College Entrance Examination Board (CEEB) were building a national system on a 200 to 400 scale, while students at the Harvard-University were classified in six divisions on a 100% scale. In the 20th-century grades became more and more standardized and colleges and universities issues grades ranging from A, B, C, D, and F to students. All those grades represented numerical values circumscribed in percent values — this grading scheme is called the „traditional"-grading scheme according to the authors.

T. Boleslavsky and C. Cotton and J. Schneider and E. Hutt [32], [34] argue that grades could also be an important determinant for latter work-performance — some companies preferred students, which performed better in their education. So, some parties interpret the grading scheme as a useful tool for a low-effort assessment even though grades could be biased or just a snapshot of a good or bad student's performance on that day. Eventually, these performances could tamper the view of the affected person positively or negatively.

Nowadays, schools, universities, and other educational institutions have the freedom to choose the desired grading scheme and involved policies themselves. A big advantage of this approach would be the adaptability to the specific curriculum and number of work packages, which do not have to be oriented to pre-specified grading schemes. Lastly, over the past few centuries the following seven main types of grading systems [35] were established globally:

1. Grading Percentage (e.g. student receives 88%)
2. Letter Grading and similar Variations (e.g. student receives a B)
3. Standard-referenced Grading (e.g. one student is compared to his classmates)
4. Mastery Grading (e.g. student is rated „masters" until reaching a pre-defined level)
5. Common Scale (e.g. student passes a course)
6. Rating of Expectations (e.g. student is compared to the desired level of quality for a specified domain)
7. Narrative Grading (e.g. a student's grades will be itemized by writing)

Assignment/Exam	Points	K1	K2	K3
Exercise 1-3	2	6		
Exercise 4 + small exam	9	9		
Exercise 5 + small exam	10	10		
Exercise 6-8	6	6	6	
Exercise 9	7	7	7	
TUWEL-Exam	20	20	20	
Practical Exam	30	30	30	
Collaboration Points	Not specified	Not specified	Not specified	

Table 2.1: Group breakdown of the IP1 course.

2.2.3 Structure and workflow

This part of the master thesis acts as a foundation for an easier comprehension of the latter chapters, which will be highly demanded due to the uncommon structure and complex grading scheme. On the bases of varying knowledge of students, the IP1 course provides three different competency-levels:

K1 stands for the competence level one, which is the initial group mainly for students with no or little experience in coding. In this group or level students can submit nine assignments, at the beginning three assignments which are designed to learn very basic concepts and this part will be mentored by a tutor or lecturer. The remaining six assignments must be prepared and submitted before the next exercise week and the solutions will be graded by lecturers. During the semester the students will be prepared to take part on a theoretical and practical exam, which can be revised once for each exam.

K2 stands for the competence level two. Students who successfully participate voluntarily on the already mentioned K2-exam can skip the first five assignments. The first five assignments are equivalent to 25 points in the IP1 course, which is also the maximum number of points in the K2-exam. K2-groups still need to participate on the theoretical and practical exam to get a positive grade for the IP1 course.

K3 stands for the competence level three. Similar to competence level two, students can participate on the so called K3-exam to skip all assignments of this lecture. The number of achievable points is 100, thus the grading is solely based on this exam and no additional exams must be taken.

The Table 2.1 provides a more detailed distribution of points.

2.3 Foundations in Statistics

In addition to the core-concepts of this master thesis, the foundation in statistics and statistical methods used in this work will be mentioned in this part. Thus, this section will not provide a complete overview of methods and processes — ultimately one should be able to understand concepts and techniques used in Section 6 and Section 7.

2.3.1 Graphs with Dual Y Axes

The authors R. Brath et al. [36] argue in their work about the usefulness of graphs with dual Y axes, which can have a number of different applications or use cases [37]. The objective is to bring two different datasets in the same chart to show the relationship between these variables. In these cases, the datasets could have different ranges or scales such as absolute and relative numbers, which can be compared by using this method. Even though previous research by P. Isenberg et al. [38] advise against using charts with dual y axes R. Brath et al. [36] found successful applications like Pareto Charts, Histograms, and Time-series Charts in their case study.

The result of this case study displays the benefits of multiple y axes on charts — however, the authors could only verify the usefulness for specific domain experts and specialists, such as data scientists and financial analysts. The major advantages affect the readability, better overview, and easier comparison of datasets. The combination of two datasets for instance allows easier local comparison to identify small differences in data as shown in the example in Figure 2.3. Also, the negative correlation and insights about the data divergence can be assessed by inverting the two axes.

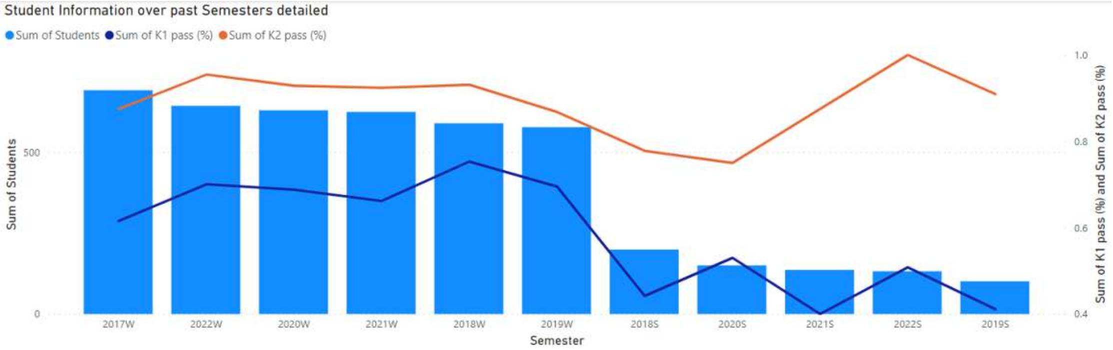


Figure 2.3: Chart with two y axes example

2.3.2 Principles of Correlation Analysis

When dealing with quantitative variables with two or more variables correlations or correlation analysis is used to work out the relation between this values [39], [40]. In order to keep the complexity of the statistical analysis manageable only linear relationships will be considered — according to N. Gogtay and U. Thatte [39] linear relationships are based on the assumption of a straight line of quantitative variables. With this relationship one can calculate a „Correlation Coefficient“, which can have values starting from -1 to

+1. Therefore, the authors discuss about three possible relationships represented by the correlation coefficient:

1. Positive correlation: This translates to a relation between variables in a positive linear manner.
2. No Correlation: When the correlation coefficient is zero, no correlation can be found within the data.
3. Negative Correlation: Analogously, this translates to a relation between variables in a negative linear manner.

These three relationships represent extremes, which are rarely reached in practical examples. However, to approximately determine the direction and strength of the coefficient, the following Figure 2.4 is used for this purpose.

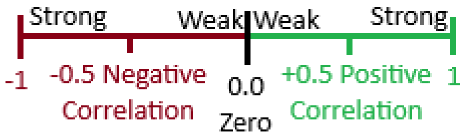


Figure 2.4: Correlation Analysis — Strengths and Directions

As within the work of C. Xiao et al. [41], this master thesis uses the Exploratory Data Analysis (EDA) approach to summarize the core characteristics and patterns of student data. The primary objective of this section is to give a short overview into the correlation analysis of different data patterns. Therefore, prior publications by A. Luxton-Reilly et al. [1] and B. Kovalerchuk et al. [36] utilize the Spearman's correlation [39], [41] to describe the pairwise relationship between two variables. According to N. Gogtay and U. Thatte [39] Spearman's correlation is based on a ranking system of the actual values, which includes the following benefits for data exploration:

- The observed data is distribution-free [41], thus does not assume a normal distribution.
- Extreme outliers have less impact on the overall result, which makes this method more robust.
- This method is effective, even if the two variables are not strictly linear.

Especially, if no prior information or analysis is conducted to new datasets the Spearman's Correlation could be a good fit to properly model for instance student data. For reference, the equation-formula for calculating the Spearman correlation will be included below. Lastly, the Figure 2.5 displays a visualization example using the Spearman's Correlation method.

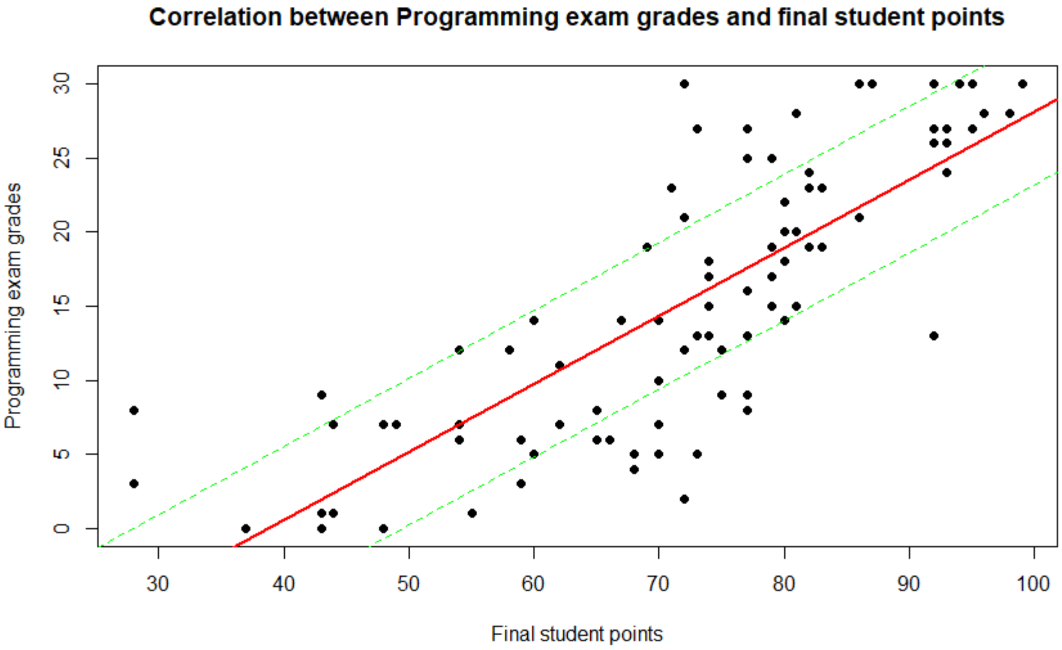


Figure 2.5: Plot example of Spearman's Correlation between two Variables

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$ [11]

- d_i shows the difference of ranks of the corresponding variables, when comparing the two sets.
- n is the total number of observation points.

2.3.3 Linear Regression

Linear regression [42] or Ordinary Least Squares (OLS) [43] is a commonly used method for the estimation of coefficients to describe the relationship between one or more independent quantitative variables and one dependent quantitative variable. A. F. Zuur

[11] https://www.simplilearn.com/tutorials/statistics-tutorial/spearmans-rank-correlation [last access: 05.03.2024]

et al. [42] argue to apply linear regression models to fit the given data in Section 7.2 by specifying these dependent and independent variables. By using linear models, the data can be represented using a straight-lined chart as shown in Figure 2.6 — generally speaking the mentioned statistical models can be used in many fields, such as biology, meteorology, and economy.

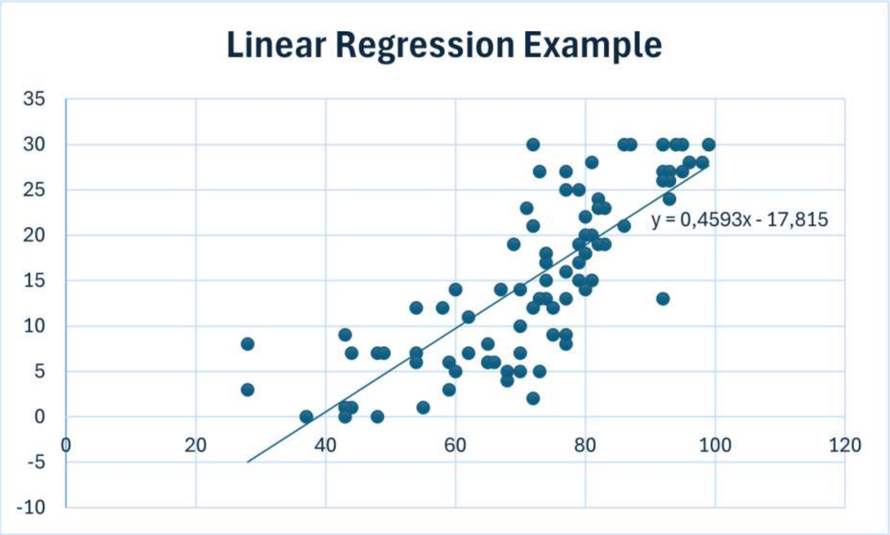


Figure 2.6: Linear Regression example

In more detail, the OLS [43] method aims to minimize the sum of squared residuals between the observed and predicted values. To put this into another perspective, the minimum sum of distances between the predicted values, which are aligned on the linear chart (see Figure 2.6), and the actual observations is calculated — the slightly adapted equation-formula by M. L. Dion [44] can be seen below.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$$

where:

$$\begin{aligned} \hat{y} &\text{ is the predicted dependent variable,} \\ \beta_0 &\text{ is the intercept,} \\ \beta_1, \beta_2, \ldots, \beta_n &\text{ are the regression coefficients,} \\ x_1, x_2, \ldots, x_n &\text{ are the independent variables,} \\ \varepsilon &\text{ is the error term.} \end{aligned}$$

The ultimate goal of these statistical techniques is to create a model, which summarizes or unifies the dataset within a few key figures and gives further insights into dependencies

P-value	**P-value %**	**Statistical evidence of possible correlation between variables**
More than 0.1	>10%	Very weak or none
Between 0.1 - 0.05	10%-5%	Weak
Between 0.05 - 0.01	5%-1%	Strong
Less than 0.01	<1%	Very strong

Table 2.2: Interpretation of the P-value on the statistical evidence of possible correlation

between the dependent variable and its features (independent variables). Due to the common usage of this statistical method, many tools such as Python (see Section 2.1.3) and RStudio[12] provide libraries such as statsmodels[13], which automatically creates a full OLS analysis of the dataset.

2.3.4 Statistical Hypothesis Testing

Before drawing conclusions of datasets as described in Sections 2.3.2 and 2.3.3, Statistical Hypothesis Tests are used to determine if the sample correlation and expected correlation are „statistically significant“ [45].

Definition 3 ***Statistically Significant:*** *To evaluate standard statistical inference of for instance two dependent variables a significance level denoted as „alpha-level“ is defined. According to C. M. Borror [46] this alpha-level describes the probability of the study rejecting the null-hypothesis assuming that the null-hypothesis is true. However, this master thesis focuses on the statistical significance on the Spearman's correlation, which uses the „p-value“ [47] to accept or reject the null-hypothesis. Values close to 1 suggest no correlation between the two variables, while values close to 0 suggest that there is a correlation in for instance student data.*

For this purpose, the p-value is used to establish evidence, that the data could indicate a statistical significance. A study by the Barcelona Field Studies Centre S.L. [47] created a table to properly interpret the p-value, when conducting a correlation analysis such as Spearman's correlation. The Table 2.2 replicates the results of this study, which will form the baseline of the latter analysis of this master thesis.

The ultimate goal of this section would be to apply an established statistical framework or method on the student data to gain additional information, which might be beneficial for both parties — students and university staff. Thus, in addition to a robust data-modelling technique also Statistical Hypothesis Testing must be conducted.

[12] `https://posit.co/download/rstudio-desktop/` [last access: 05.03.2024]

[13] `https://pypi.org/project/sweetviz/` [last access: 05.03.2024]

CHAPTER 3

State of the Art

The following chapter of this master thesis represents the academic state of research in the areas of EDM, Computer and Information Science Education (CSE), and Data Visualization. Thus, this section gives a brief overview of the current state of research in the development of measures and techniques to improve or support education for computer science students. Which includes the currently used tools and workflows to manage and visualize student information and possibly give valuable insights to estimate the student's future performance. The research of this master thesis focuses mainly on methods, which do not require much extra activity by lecturers, tutors, or instructors.

3.1 Current State of Research

As mentioned above, this master thesis focuses firstly on basics and current developments in EDM to get a brief overview of available techniques and well-established concepts in this domain. These concepts are not just limited to European universities or faculties. Secondly, this section aims on modern methods to visualize information or data effectively and lastly create a distinction between the collected and established practices and the findings from this master thesis.

3.1.1 Educational Data Mining

In the past few years, generating, collecting, and searching data has increased dramatically [48], which enlarged the number of possible applications, use cases, and concepts for Data Mining like real estate, customer relationship, social media, healthcare, etc. [7], [10], [49]. Among these new applications of Data Mining lies also EDM, which is an emerging domain dealing with education-related data and examining students' performance and learning capabilities. According to S. K. Mohamad and Z. Tasir [50], E. B. Costa et al. [51] and K. Bunkar et al. [10] the purpose of Data Mining is to observe previous student performance and examine trends, which contributes to educational research. Additionally, gaining further insights in educational settings [50] and assessing indicators of student latter performance [12].

The Data Mining Task and used platform varies across previous studies, which according to S. K. Mohamad and Z. Tasir [50] depends on the ultimate objective. Interestingly, a majority of studies focus either on external systems [52]–[54], online tracking, and learning [55]–[57], or external environments [58], [59] to find patterns or correlations in the given data. Based on this data, assumptions could be created or the education processes may be improved. Most of these mentioned publications were sourced from conferences or journals by the following Data Mining Tasks for EDM:

- Sequential Patterns: To find sequential patterns, student teams were assigned to use Trac for any online interaction and collaboration. By using this method J. Kay et al. [52] were able to extract sequences and common patterns and events in their learning behavior.
- Association Analysis: For the association analysis M. Pechenizkiy et al. [56] used association rules to predict student's behavior. Further indicating if and when questions are answered correctly or which factors or characteristics are important to pass or fail exams.
- Clustering: After preprocessing student data, L. Talavera and E. Gaudioso [53] used a model-based clustering approach for both continuous and discrete datasets. Firstly, creating cluster profiles, which are then categorized according to certain discriminants. Likewise D. Perera et al. [54] tried to develop an assistive model by using another clustering approach, which consists of multiple attributes for the identification of similar observations or groups.
- Classification: As described by A. Anjewierden et al. [60] in classifications one is applying characteristics of a model as certain classifiers and afterwards comparing the classifier with another model feature. In general, classifications may provide valuable insights and information, which can be used by domain-experts to improve for instance learning behavior of students or to increase the effectiveness of study-groups and classes.

Definition 4 ***Trac:*** *Trac*[14] *is an open-source issue tracking system, which contains additional features for management and development projects [54]. The tool offers a timeline and roadmap to show project and event histories and upcoming tasks, which simplifies the management of projects, while at the same time providing tracking and monitoring features. Lastly, Trac shares an open group wiki for efficient collaboration between managers, developers, and other users.*

[14] `https://trac.edgewall.org/` [last access: 04.01.2024]

More recent papers, articles, and other publications deal mainly with classification of important attributes for student performance [10], and effectiveness of different EDM methods. Also, techniques for the indication of computer science students' academic performance [51], and employment of early assessment tasks [12], [61] or Micro-learning units [62] were developed to predict if students are at risk of failing programming courses. Further research is concerned with certain machine learning techniques [9] which are based on pre-specified classifiers to examine the usefulness of machine learning models to prevent student dropouts in the Distance-Learning environment.

Definition 5 ***Micro-learning:*** *According to J. Skalka and M Drlik Micro-learning [62] is a specific didactic technique, which uses digital media to offer small and self-contained information — these „learning bits" are used for short learning activities. The invention of such compact e-learning modules*[15] *is steadily emerging due to the expansion of the mobile market and use of smartphones or any other portable network devices.*

Additional findings by A. A. Mubarak et al. [63] proposed to use and compare Logistic Regressions and Input-Output Hidden Markov Models to alternative machine learning techniques, which were able to issue predictions with up to 84% accuracy.

Lastly, according to more recent literature increased research and interest is assigned to predict students at risk of dropping out in introductory programming courses by self-efficacy and well-established prediction models [7]. On top of that, S. N. Liao et al. [5] and C. Gordon et al. [4] are developing lightweight early prediction models to determine and indicate when and if students will be doing poorly in computer science courses.

3.1.2 Data Visualization

Data visualization enables a convenient and informative view of the initial or preprocessed data, which can be further used by domain-experts, such as lecturers, tutors, and instructors to improve the decision-making process [64]. Similar to this description, S. Gama and D. Goncalves [65] argue that Data Mining techniques might be complex to comprehend, analyze and further assess. Creating for instance multi-level visualizations could overcome these limitations and give a manageable overview of the data over a certain period of time.

Some approaches and tools focus on establishing interactive visualizations and environments by increasing usability and visual response-time to any kind of changes in the dataset or the underlying models. According to T. Soukup and I. Davidson [66] effective visualization of data often requires project planning and additional data preparation steps to relate, transform, and verify the dataset.

[15] `https://www.edume.com/blog/what-is-microlearning` [last access: 04.01.2024]

S. Slater et al. [67] argue that EDM has also increased the demand for visualization approaches — the ultimate goal is to gain additional knowledge by discovering new features and coherences in the domain of education and student's learning performance. Interestingly, R. Paiva et al. [68] discuss about different characteristics of data visualizations in the EDM-domain such as utility, usability, aesthetics, color schemes, etc. — these features might ultimately influence the readability of certain visualization, which must be considered when choosing and configuring data visualization tools.

As stated in the article by A. Goncalves et al. [69] the number of tools, which also offer learning analytics and visual representations is increasing steadily. The authors refer to already established management platforms, such as the Moodle-environment [15], which provides various visualizations and graphs to get an overview of the performance, activities, and overall points in courses. However, the researchers used certain Data Mining and visualizations tools to generate sample results from undergraduate students and thus analyze their behavior within a certain course. Little research was conducted by considering these management platforms and possible capabilities or synergies, when combining all three topics EDM, data visualizations and the Management Platform Moodle.

3.2 Available Tools

To complete the overview of the current research, this master thesis provides well-established tools or software programs, which are used either in the education or the data visualization-domain. Additionally, the Management Platform TUWEL and its external webservices will be explained in more detail to gain basic knowledge about the latter data mining process.

3.2.1 Learning- and Management Tools

The following tools are developed to support students, teachers, and instructors by sharing learning content and further visualize student behavior. In other words, these LMSs [70] simplify and support the teaching process, while at the same time offering a platform to discuss, submit, and conduct online exams. Nevertheless, this master thesis will focus on tools mainly used in Higher Education Organizations (HEOs) [71], especially due to the high number of participating parties. Lastly, some alternatives will be included to get a brief perspective on additional processes and tools.

Blackboard LMS

According to I. Dobre [71] and T. I. Tawalbeh [72] one of the most widely used LMS for academic institutions is the Blackboard Learning Management System (Blackboard LMS)[16], which basically provides various online course management features. Lecturers,

[16] https://www.anthology.com/ [last access: 07.01.2024]

tutors, and instructors are able to manage whole courses, by posting announcements, assignment- and exam grades, conduct online quizzes and exams, etc. Blackboard LMS[17] simplifies the communication between academic staff and students and even offers developers to establish external applications by using the provided REST APIs [73].

However, C. Hall [74] asserts that the quality of the implementation is pivotal for the effectiveness and support of this commercial learning platform. Thus, the preparation and planning phase for certain courses and the setup of the foundation might take up comparably more resources.

Remark: Since October of 2021 the name of Blackboard LMS was changed to Anthology[18] — within this master thesis both terms will be used to match the academic literature and information provided by recent websites.

Sakai

Sakai[19] is an open-source LMS [75], which has an improved GUI and includes automatic markers, increasing the usability and reducing the management overhead. According to a study conducted by S. Dube and E. Scott [76] Sakai requires initial training to use the learning platform properly. Nevertheless, the authors argue that Sakai turns students into passive consumers of its content, experiences, and it helps to transmit course information for academic purposes. The system was adjusted and developed for academic staff and students for efficient in-person and online experience and participation.

eFront LMS

The eFront[20] LMS is mainly configured for enterprises or larger organizations and provides support in complex learning environments. Z. Fariha and A. Zuriyati [77] listed the supported features, which includes digital libraries, chats, forums, wikis, messages, etc. The commercial versions of the mentioned LMS has integrated modified features to further increase the learning process online and in-person. However, only the free open community version uses an Open Source Initiative (OSI)[21] accepted license.

DigitalChalk

In comparison to the previously mentioned LMS DigitalChalk[22] was developed by Sciolytix as a cloud-based LMS in combination with Cloud Computing features to distribute study-

[17] https://www.anthology.com/ [last access: 07.01.2024]

[18] https://www.anthology.com/news/anthology-completes-merger-with-blackboard-launches-next-chapter-in-edtech [last access: 07.01.2024]

[19] https://www.sakailms.org/ [last access: 07.01.2024]

[20] https://www.efrontlearning.com/ [last access: 08.01.2024]

[21] https://opensource.org/ [last access: 08.01.2024]

[22] https://digitalchalk.uk/?utm_source=google.com&utm_medium=organic [last access: 08.01.2024]

content to all stakeholders at any time [71]. According to J. B. Idoko and J. Palmer [78] DigitalChalk can also be used as an online training software by sharing or selling online courses. Due to its cloud-based implementation, the software can be efficiently used for all kinds of enterprises and exhibits a high level of scalability. The authors argue that using the LMS includes several limitations or other obstacles which must be considered when using this learning platform.

Moodle

Among the most popular LMSs stands the Moodle[23] environment, which is widely used as a teaching and learning hub for higher education [79] by educational stakeholders and even software developers or researchers. It provides a well-established e-learning kit whether for in-person or online lecturing. S. Gamage et al. emphasize the importance of LMS such as Moodle during the COVID-19 pandemic, which limited face-to-face interaction between students and university staff.

In general, Moodle is an open-source software, which can be installed and configured by any user for testing or simply using its features, external APIs, and more advanced features like course analytics. Additionally, it provides its services in various languages and can be configured for any kind of courses, while at the time offering an user-friendly environment. Universities such as the TU Wien have developed their own Moodle-instance called TUWEL, which implemented advanced features for analytics and communication. One of TUWEL's features is an extension by the Universität Münster[24], which enables to create forums anonymously. A recent study by B. Oguguo et al. [80] even recommends the use of the LMS Moodle, which outperformed other learning platforms.

3.2.2 Data Visualization Tools

To further analyze, represent, and subsequently interpret large amounts of student data, this master thesis will list methods and data visualization tools accordingly. S. Ajibade and A. Adediran [64] refer in their journal article about the different forms or graphs to visualize data — among these methods or techniques are charts, tables, plot diagram timelines, data series, trees, and networks. Depending on the chosen visualization technique, values or data-attributes can be summarized more effectively.

Some of the developed tools can also be used for Business Intelligence (BI) use cases by providing a high level of usability, interactive design, and external APIs to other software programs or files.

[23] https://moodle.org/ [last access: 08.01.2024]

[24] https://www.uni-muenster.de/Kowi/forschen/index.html [last access: 08.01.2024]

Tableau

Tableau[25] became more relevant during the emergence of the BI industry [81] — organizations searched for ways to visualize or report large amounts of data to conduct further analysis and support the decision-making process in the management department. Therefore, according to D. G. Murray Tableau offers an unique desktop design and visual analysis tools for structuring datasets and creating informative dashboards. The software program provides a Personal Edition and a Professional Edition, which includes additional features and options to source data. Additionally, Tableau has established a platform to share content and collaborate with others.

The software is based on workbooks, which contain worksheets, different measures, data planes, pre-defined charts and options, and many more features to customize the data visualization as desired. An exemplary visualization can be viewed in Figure 3.1 showing the mentioned measures and visualization components.

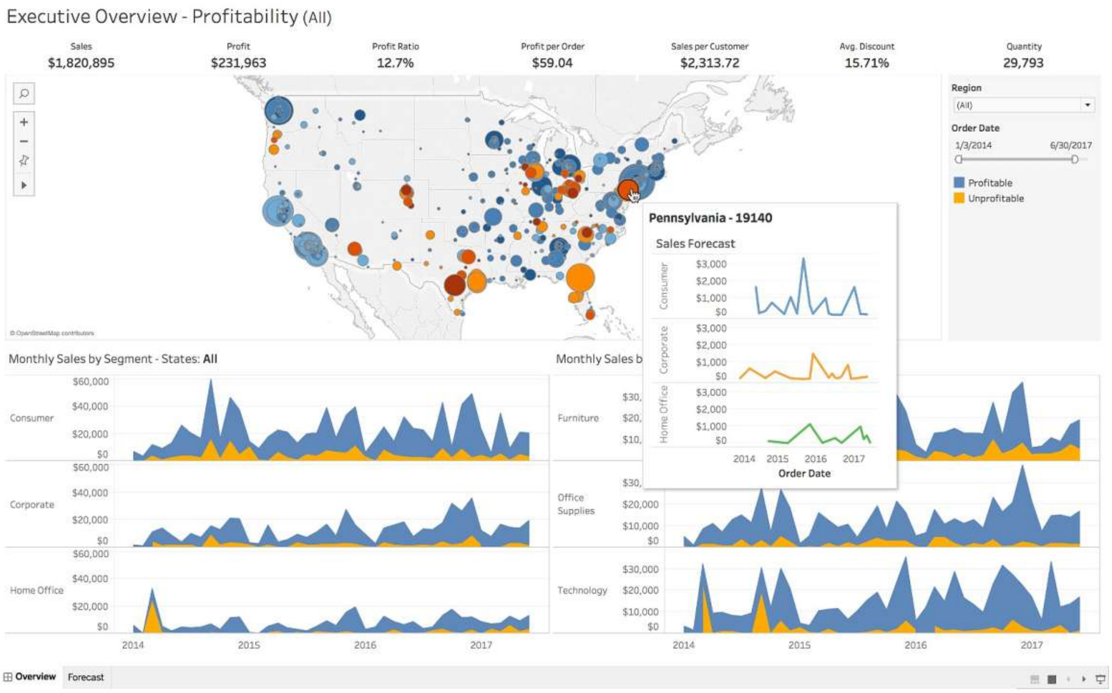


Figure 3.1: Tableau Dashboard example view[26]

[25] https://www.tableau.com/ [last access: 08.01.2024]

[26] https://cdns.tblsft.com/sites/default/files/pages/platform-3-800x5002x.jpg [last access: 08.01.2024]

Google Data Studio

In recent years, Google Data Studio was developed as part of Google's Analytics360 suite, which provides extensive tools for visualizing complex datasets. According to G. Snipes [82] the main purpose of this tool is to analyze and interpret social media information and to conduct projects for web analytics. It supports external Database tools and is compatible with Google Sheets[27], which enables to input various types of data in information provided in different domains.

Information or data is represented on dashboards, which can be customized with a range of pre-defined graphs, charts, maps, average values, etc. as shown in Figure 3.2. A big advantage is also the possibility to export and link Google Data Studio with other applications by Google LLC[28].

Remark: Since October of 2022 Google announced a rebranding of Google's Data Studio to Looker Studio[29] — however, within this master thesis both terms will be used to match the academic literature and information provided by recent websites.

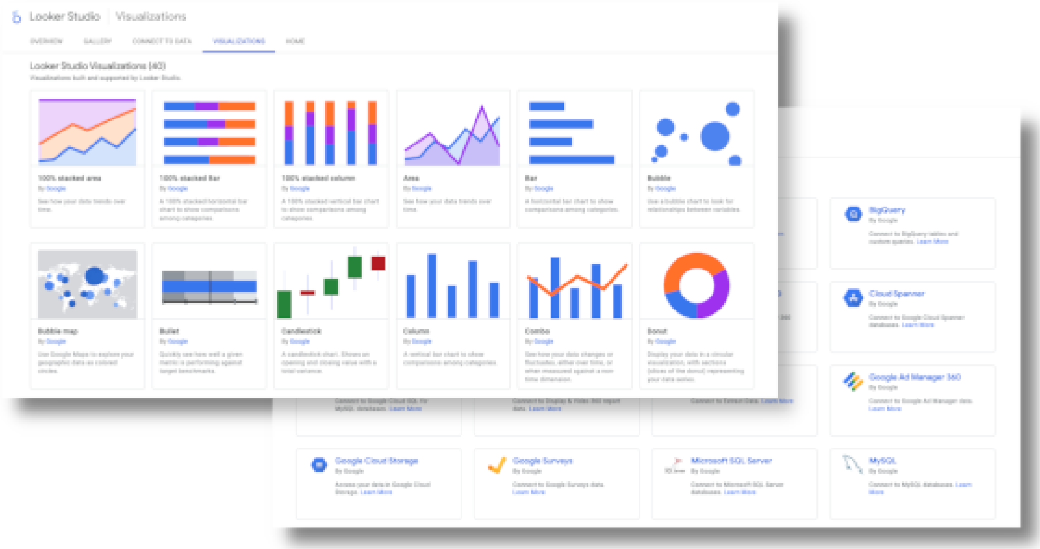


Figure 3.2: Google Data Studio Dashboard example view[30]

[27] https://www.google.com/sheets/about/ [last access: 09.01.2024]

[28] https://about.google/products/ [last access: 09.01.2024]

[29] https://cloud.google.com/looker-studio [last access: 09.01.2024]

[30] https://developers.google.com/static/looker-studio/images/looker-studio.png?hl=de [last access: 09.01.2024]

Apache Superset

Apache Superset[31] is an open-source visualization platform, which can also be used for BI use cases or exploring large amounts of data using highly customizable dashboards [83] (see Figure 3.3). These dashboards can be configured using graphs, averages, maps, and other visual components to create a manageable and structured overview of the fetched data. S. Shekhar [84] states that Apache Superset is a very interactive platform, which does not require any programming knowledge to analyze and process datasets. The software platform focuses on an user-friendly design and enables users to collaborate and share their work to other contributors or viewers.

According to its website[32], Apache Superset also provides features for more advanced stakeholders to manipulate or create Structured Query Language (SQL)-queries or edit datasets, while at the same time supporting a variety of databases [85] and external tools. Lastly, the deployment of the software can be done using a Docker Engine[33] or with Kubernetes[34] — thus, Apache Superset provides extensive documentation and support for starters.

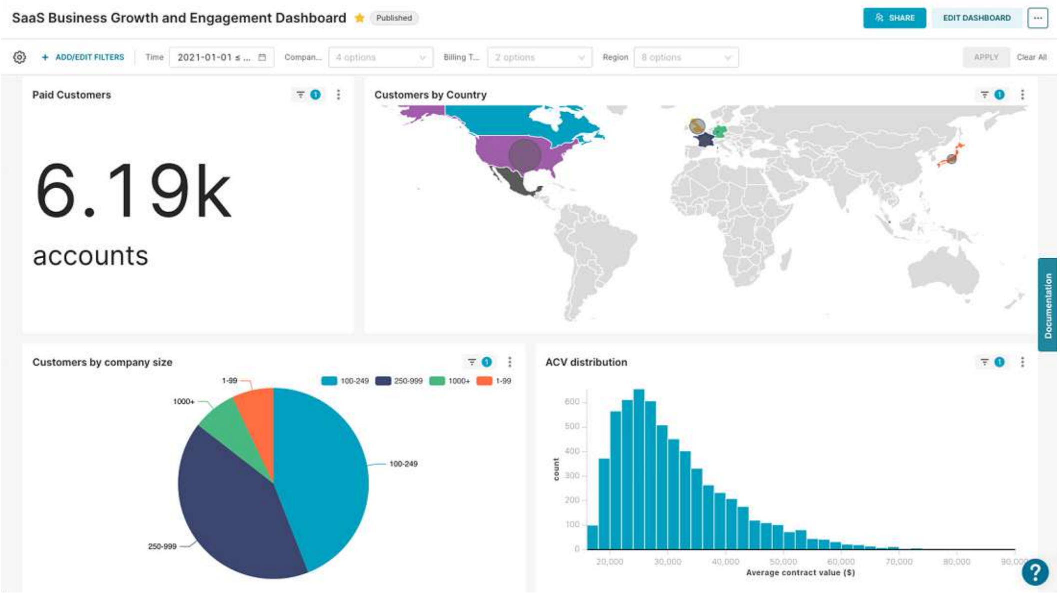


Figure 3.3: Apache Superset Dashboard example view[35]

[31] https://superset.apache.org/ [last access: 09.01.2024]

[32] https://superset.apache.org [last access: 09.01.2024]

[33] https://superset.apache.org/docs/installation/installing-superset-using-docker-compose/ [last access: 09.01.2024]

[34] https://superset.apache.org/docs/installation/running-on-kubernetes/ [last access: 09.01.2024]

[35] https://superset.apache.org/img/hero-screenshot.jpg [last access: 09.01.2024]

Microsoft Power BI

Microsoft Power BI[36] is an interactive software for data visualization, which primarily focuses on the business intelligence domain. In general, Power BI is an advancement of the three add-ins in Microsoft Excel[37] *Power Pivot*, *Power Query*, and *Power View* [86]. Currently, Power BI is an independent tool, which does not require any other Microsoft Office software — it combines simplicity, user-friendly design, different visualizations, and interactive capabilities. These features satisfy a wide range of stakeholders for creating reports or conducting visual analysis. Nevertheless, Power BI still provides support for many queries, files, and various software programs to import data.

Using Power BI does not require any programming knowledge, while at the same time supporting many tools and representations. These aspects make Power BI a very powerful tool for analyzing organizations, processes, and machine data, which amplifies data insights and establishes an informed decision-making process. Tutorials and various guides[38] help inexperienced users to utilize some of its more advanced features to manipulate datasets before visualizing the data in form of dashboard-pages. Microsoft offers five versions of Power BI[39] — each of these versions are optimized for different applications or use cases. For the purpose of this master thesis Microsoft Power BI Desktop[40] will be used to explore the data and create a visual models by using all three previously mentioned add-ins as shown in Figure 3.4.

3.3 Distinction from Current Literature

This master thesis presents a brief overview of the research area, available tools, and current development in the EDM-domain based on a semi-systematic literature review following H. Snyder's approach [87]. One of the outcomes of this research was that there is no comparable solution or model considering the scope and ultimate goal of this master thesis. Generally speaking, the research field of EDM is still emerging, especially tools or models for the identification of student's performance [4], [5], [12], [13] are quite new and require powerful BI applications, visualization tools, and Artificial Intelligence (AI).

Ultimately, this master thesis intends to contribute as a foundation for future work regarding EDM, while using a LMS such as the Management Platform Moodle. In more detail, Moodle already provides a number of ways to analyze and visualize student data, assignments, grades, etc. However, after consultation with Dr. S. Podlipnig and

[36] `https://www.microsoft.com/en-us/power-platform/products/power-bi` [last access: 12.01.2024]

[37] `https://www.microsoft.com/en-us/microsoft-365/excel` [last access: 12.01.2024]

[38] `https://learn.microsoft.com/en-us/power-bi/` [last access: 12.01.2024]

[39] `https://powerbi.microsoft.com/en-us/downloads/` [last access: 12.01.2024]

[40] `https://powerbi.microsoft.com/en-us/desktop/` [last access: 12.01.2024]

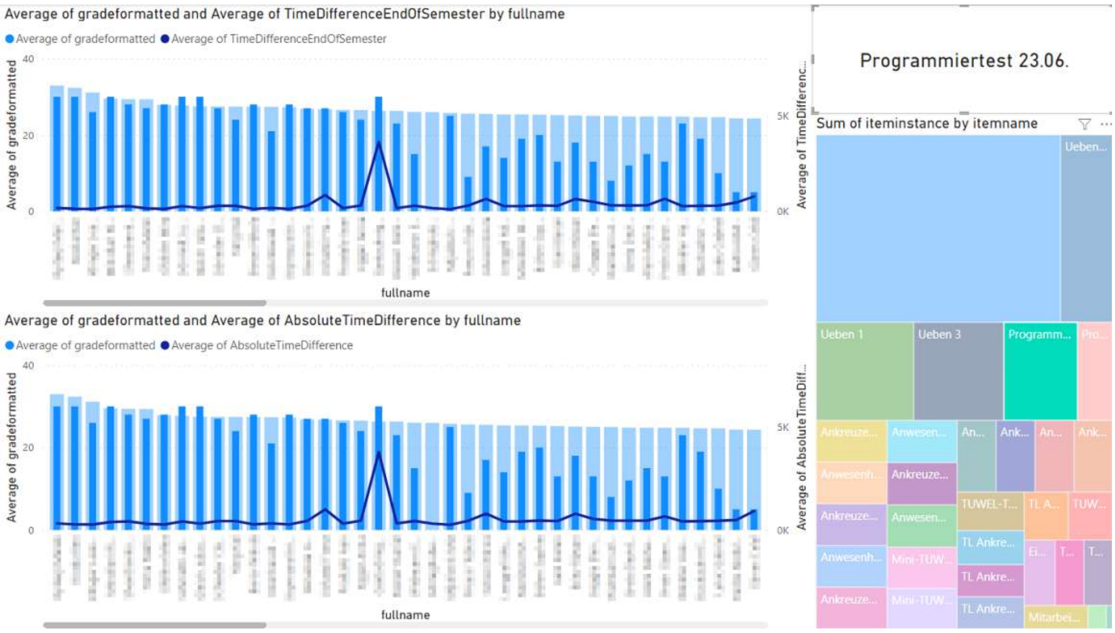

Average of gradeformatted and Average of TimeDifferenceEndOfSemester by fullname
● Average of gradeformatted ● Average of TimeDifferenceEndOfSemester
Average of gradeformatted
40
20
0
5K
0K
Average of TimeDifferenc...
fullname
Average of gradeformatted and Average of AbsoluteTimeDifference by fullname
● Average of gradeformatted ● Average of AbsoluteTimeDifference
Average of gradeformatted
40
20
0
5K
0K
Average of AbsoluteTimeDiff...
fullname
Programmiertest 23.06.
Sum of iteminstance by itemname
Ueben...
Ueben 1
Ueben 3
Programm...
Pro...
Ankreuze...
Anwesen...
An...
Ank...
An...
Ank...
Anwesenh...
Ankreuze...
TUWEL-T...
TL A...
TUW...
Ankreuze...
Anwesen...
TL Ankre...
Ei...
T...
T...
Anwesenh...
Mini-TUW...
TL Ankre...
Ankreuze...
Mini-TUW...
TL Ankre...
Mitarbei...

Figure 3.4: Power BI example report view

colleagues, who are mainly responsible for the IP1 course of the TU Wien the following feedback was extracted during a short interview session:

- TUWEL lacks an activity profile of all students, to better understand and compare student behavior. This activity profile summarizes the student activities such as submitting assignments or commenting on a forum post.
- TUWEL does not provide a summary of all student grades over the whole course to compare students visually.
- TUWEL does not give further quantitative insights into discussions, posts, forums.
- TUWEL provides only rigid reports and visualizations, which are in some cases coarse-grained or inadequate. There is no possibility to change the view or granularity according to certain use cases.

As mentioned above, the feedback is related to TUWEL, which is a Moodle-instance managed, developed, and extended by the TUWEL-support team using various Moodle-packages by different vendors like the Universität Münster[41]. Thus, currently no system, model, or attempt was made to further analyze and interpret student information in TUWEL — exploring this system and sharing the capabilities could be a contribution to

[41] https://www.uni-muenster.de/de/ [last access: 12.01.2024]

the EDM-community and future work in this research area.

It is planned to include a correlation analysis based on the results of the planned case study on the developed visualization and data model. This analysis could detect important activities or student behavior within a course to achieve a better grade and ultimately pass an introductory programming course. Examples for this would be submission of assignments or effort in example TUWEL-quizzes. Additionally, the proposed prototype or model helps tutors, lecturers, and instructors to understand student behavior and possibly identify common patterns, which might lead to similar results. This would not only improve the decision-making process, but also highlight outliers or students who might need more attention.

Nonetheless, there are already some established models, which try to identify or predict student performance — C. Gordon et al. [4] talk about common techniques such as conducting surveys, collecting demographic information, and including questionnaires. The approach by S. Liao et al. [5] aim to combine a set of elements or models to create accurate predictions and decrease biases. A. Veerasamy et al. [12] use non-machine learning models to undergo formative assessment of student's performance, engagement, and ultimately results.

In the end, little research is published to indicate student performance using the Moodle platform, which is based on different data structures and provides only limited access and information over its External Services [25]. This master thesis tries to mitigate these issues and offer some alternative ways to indicate or visualize student performance, while maintaining the tool's lightweight profile by refraining from the implementation of supplementary monitoring or tracking systems.

CHAPTER 4

Methodology

The aim of this master thesis is to design a sustainable and customizable visualization tool and statistical model. This prototype will be essential for university staff to allocate available resources efficiently, while at the same time keeping track of all students participating in a programming course. The methodology is divided into two separate parts. Firstly, conducting a semi-systematic literature review, to get a brief overview of available LMS, monitoring, as well as data visualization tools.

Secondly, the practical part of this master thesis will follow a design-science approach, which creates a software-artifact to gain further insights into an introductory programming course. In order to test this model, a case study on the IP1 course will be conducted — the results and insights will be reflected and assessed afterwards. To sum it up, this chapter describes the methodological approach and activities in more detail to better understand the scientific approach and added value for the mentioned research areas, which might benefit various stakeholders.

4.1 Research Questions

This section pertains to the research questions previously mentioned in Section 1.1 and elucidates the planned methodology to address the specified research questions accordingly. Most of the following research questions will be answered using both — the Semi-systematic Literature Review [87] and Design Science approach by P. Offermann et al. [88].

RQ1 : „To what extend can students at risk of failing a course be determined by previous activities?“

O1 : In order to answer this question all fundamental parts of grading and grading schemes will be researched beforehand. This includes commonly used systems, models, and insights in other academic courses fitting the research domain. Based on this information the developed model will be used to categorize and assess the case study, which is followed by a correlation analysis and summary of the gained insights.

RQ2 : „Which combination of attributes is most relevant to approximate a student's final score?"

O2 : This part is closely related to RQ1 by filtering important activities or student behavior to efficiently pass programming courses in theory. Thus, it is planned to conduct short interview sessions with domain experts, which are in our case tutors, instructors, and especially lecturers. The ultimate goal is to create a minimum set of activities, which are latter validated using the case study on the IP1 course.

RQ3 : „Which attributes or properties of course-contents or taught programming skills are hard to comprehend for computer science students?"

O3 : This research question will be examined by reviewing the current work, assessment of domain-experts and comparing the results with the visualizations or data based on the mentioned case study.

4.2 Semi-systematic Literature Review

One of the first steps of this master thesis will be to conduct a review of current research and available models. H. Synder [87] distinguishes between different types of literature reviews. However, these approaches include a quantitative, qualitative, or mixed design during different phases of the literature review. Thus, the author refers to the main three types as *systematic review*, *semi-systematic review*, and *integrative review*. Considering all possible types a semi-systematic review combined with a quantitative design approach would be expedient. Therefore, this thesis will not focus on all accessible research, but rather on selected terms and criteria regarding efficiency, usability, and indicating models or tools.

The approach according to H. Synder [87] consists of four stages to conduct a semi-systematic literature review.

1. **Design:** The first step before starting to write a literature review is to look for meaningfulness and level of contribution of one own's research. Basically, this phase of research tries to find the actual approach of the literature review and tries to fill in the gaps of the affected area. Additionally, the purpose or contribution of the

project can be estimated and inclusion/exclusion criteria will be chosen in the next steps.

2. **Conduct:** According to H. Synder [87] creating a pilot experiment of the literature review process would help to identify and improve inclusion/exclusion criteria such as search terms. This process is improving iteratively to ensure a reliable and qualitative search protocol. In general, this phase represents the data extraction phase and collection of knowledge of the filtered articles related to „at risk students" in programming courses, lectures, or trainings. For the information gathering, only the established academic search service Google Scholar[42] will be used — further narrowing down research to English or German.

3. **Analyze:** In the case of this master thesis, it will be valuable to evaluate important attributes or aspects for indicating student's performance manually. The fetched data will be collected, compared, and analyzed towards the research goals.

4. **Writing:** For the last part of the literature review the underlying motivation and need for research must be communicated to readers. Lastly, this includes a distinction from the current literature in Section 3.3 by explaining which research areas or topics have no or just little research — this master thesis aims to answer these sub-domains as well as possible. Ultimately, a major objective would also be to focus on the reporting of qualitative reviews and evaluating results of interventions.

4.2.1 Technology Review

Prior to starting the pre-processing process and latter visualization of data, several tools and techniques were examined in Section 3.2 following process-steps in Section 2.1.3. The functionality and value of the developed model will be assessed based on the following questions, which represent requirements of different stakeholders — this list was created in cooperation with Dr. S. Podlipnig.

1. Is it possible for students, lecturers, and tutors to use the software technology for free? Which licenses are required to use this software in general?

2. Is the tool robust when changes occur and compatible with other systems?

3. Is the tool properly maintained and further optimized and supported by the software tool providers?

4. Is proper documentation available and can this tool be used intuitively?

5. Can the tool be edited, extended, or customized for new use cases?

[42] `https://scholar.google.com/` [last access: 12.11.2023]

In the end, after comparing a few external tools, Microsoft Power BI[43] was used to implement the data pre-processing and latter visualization. One of the biggest advantages of Power BI is the extensive documentation, videos, and guides by Microsoft and the continuous support by the company. As mentioned in Section 3.2.2, the visualizations, patterns, graphs, labels, etc. are highly customizable, while at the same time providing a high level of usability and intuitive design. Changes in the lecture's content such as increase of assignments would not significantly affect the visualizations, due to the generic design of the BI-tool.

4.3 Design Science Approach

In general, the design science approach consists of creating a model or tool, conducting an experiment based on this model and afterwards evaluating the findings. However, multiple approaches were described in the previous literature, such as *soft design science* [89], *design science for information systems* [90], *detailed research for design science* [88], [91]. Considering the purpose of the master thesis, selecting the research approach by P. Offermann et al. [88] would consist of the three main steps of every design science methodology.

1. **Problem Identification and motivation:** Currently, instructors or tutors often do not own a model or tool to identify students at risk of failing the programming course. The underlying problem is about the uncertainty how to effectively allocate resources to support students in their studies [4], [5], [49]. Furthermore, reviewing the state-of-the-art approaches and best practices could serve as a solid foundation to tackle the addressed issue. Pre-evaluation of the relevance of the problem will be conducted, thus compiling a research hypothesis to exhibit a link between the problem and the solution space and further describe advantages of the approach. Lastly, defining objectives, measures, or specific goals to determine the effectiveness of the model or approach.

2. **Design and development of the solution:** The first step, before inspecting the data is to understand the project objectives and involved requirements. This section also includes all processes to clean up, unify, and filter the raw data. The goal would be to rearrange or modify the given data to be used for the development of a model without losing valuable information. Additionally, several modelling techniques and ways of representation will be considered. This involves calibrating optimal parameters and rearranging attributes or values to construct an effective statistical model.

3. **Evaluation of the model:** In the case that the predefined research hypothesis cannot be evaluated as a whole, the scope of the hypothesis could be further

[43] https://www.microsoft.com/en-us/power-platform/products/power-bi [last access: 12.01.2024]

narrowed down. A practical experiment or test is then used to validate the (adjusted) hypothesis. This means, qualitative and quantitative metrics will be used to test the performance of the developed model on courses in previous semesters. One of the final steps is the evaluation process of the conducted experiment. Hence, comparing the results of the experiment with the predefined goals and determining the level of contribution to tackle the mentioned problem. Lastly, including a summary of the results, findings and insights and possibly extending the developed model to enlarge the area of application - not only limiting the model for computer science courses.

4.4 Proof of Concept

The proof of concept is based on the steps defined in Section 2.1.3, by fetching course information using Moodle's External Services and API. Jupyter Notebook [28] will be used to transform and save the data in CSV-files. These files are then pre-processed, sorted, and then visualized by Power BI — initially, this step is highly customizable, which is beneficial for changing requirements or structures in lectures. Ultimately, the proof of concept will be realized using the IP1 course to visualize and statistically model student data, such as grades, assignments, forum-activities, exams, etc. The information gained using this approach supports the decision-making process for student staff and enables efficient allocation of resources.

4.5 Evaluation and Assessment

Lastly, the evaluation of developed tool will be conducted by aggregating a quantitative analysis and an expert evaluation to compare or even complement the statistical findings. The first evaluation step would be to create a statistical model [56] of the data by following existing best practices and approaches in previous literature [4], [7], [61]. The data from this analysis might bring beneficial information to scrutinize current processes and improve the overall resource allocation for affected courses.

The second evaluation step considers all available TUWEL modules, which might impact the final student grade directly or indirectly. Thus, a questionnaire with domain experts will be conducted, which represents the current assessment of external and internal influences during the IP1 course. Afterwards, a final comparison between these evaluation methods could involve valuable results, which can be used to further optimize the management of courses at the TU Wien.

CHAPTER 5

Extract, Transform, Load Implementation

In the following chapter of this master thesis the development process of the designed tool will be described in more detail. Thus, for implementing the visualizations the Extraction-Transformation-Loading (ETL) approach [92] was used to fetch and transform the raw data from TUWEL's API documentation[44], which originates from Moodle's External Services (Section 3.3). Afterwards, the transformed and cleaned data tables will be loaded, linked, and visualized using the BI tool Power BI as described in 4.4.

5.1 Data Extraction and Structure

This sections explains the data extraction process in more detail, which is not documented publicly. Thus, this master thesis tries to fill in the gaps to ensure reproducibility and a high level of transparency into the development of the tool and the development process and previous obstacles.

5.1.1 Pre-requisites for Data Extraction

Prior to extracting the data, some research and consultation by the technical support team of TUWEL must be concluded, due to the rather uncommon approach to download course content or student data. One of the pre-requisites to use the TUWEL API is to obtain the Moodle-Token in the network tab, while being logged into the TUWEL user account. For this purpose, the TUWEL consultant Dr. G. Rakoczi prepared a step-by-step guide to get and insert the token into the actual API request. The following enumeration of steps was adapted and shortened to coincide with the requirements of this master thesis and to increase the readability:

[44] https://tuwel.tuwien.ac.at/webservice/wsdoc.php?id=4847 [last access: 12.01.2024]

1. Open the monitoring or network view of the desired browser (preferably Google Chrome[45]).

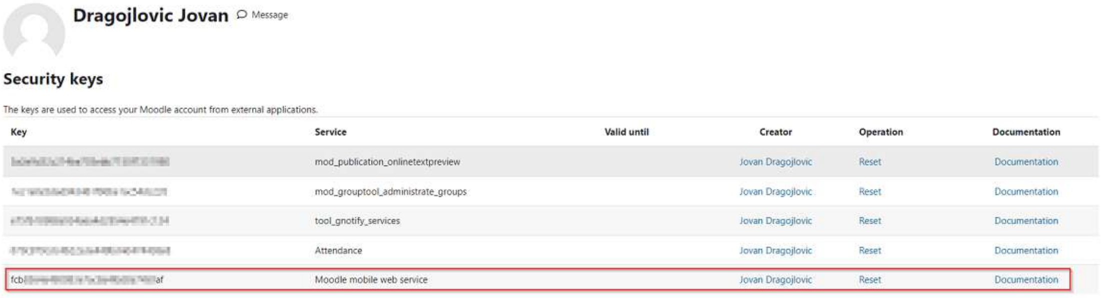
2. Call the URL `https://tuwel.tuwien.ac.at/admin/tool/mobile/launch.php?service=moodle_mobile_app&passport=<password>` by providing a strong password in the brackets „*password*“. For this step the TUWEL user must be already logged in.

3. The response should contain the following structure „*moodlemobile://token=<token>*“ — for example „*moodlemobile://token=NGNhTURlOTc2MLGlNjYzMDBiZTUwZjlmN2Q2ZGMxYjg6OjpmM2I4MzY0YTQ5MDA4M2U3YWM4HATmZGQ4YTc0NDDhZjo6OklXemgxalUxd2puCZDxSzlwQ2xqamZuOGlMUVM5TktHM1JUZDlVcVlEbThqSE5pejAySE41UlE5Q1ZxM0tvTDI=*“.

   However, this token is encoded in a base64 format, which must be decoded beforehand by using for instance an online decoder[46] or copying the security key of Moodle's Web Services from TUWEL's user account as seen in Figure 5.1[47].

   As additional information: In order to extend the lifetime of this key, seek consultation with the TUWEL technical support team or join the Distance Learning Online Office Hours[48].

4. When the key is directly decoded only the middle part between triple „:“ contains the TUWEL security key — the decoded text should look similar to this „*4ca96e9760de66300be 50f9f7d6dc1b8:::<token>:::IVzh1jU1wjnUCqK9pCljjfn8iLQS9NKG3RTd9UqYDm8 jHNiz02HN5RQ9CVq3KoL2*“ and the specified token can be extracted.



Dragojlovic Jovan

Security keys

The keys are used to access your Moodle account from external applications.

Key	Service	Valid until	Creator	Operation	Documentation
[redacted]	mod_publication_onlinetextpreview		Jovan Dragojlovic	Reset	Documentation
[redacted]	mod_grouptool_administrate_groups		Jovan Dragojlovic	Reset	Documentation
[redacted]	tool_gnotify_services		Jovan Dragojlovic	Reset	Documentation
[redacted]	Attendance		Jovan Dragojlovic	Reset	Documentation
fcb[redacted]af	Moodle mobile web service		Jovan Dragojlovic	Reset	Documentation

Figure 5.1: TUWEL Security Keys example

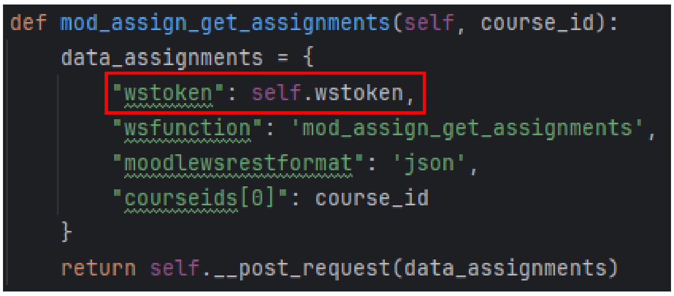
[45] `https://www.google.com/chrome/` [last access: 12.01.2024]

[46] `https://www.base64decode.org/` [last access: 18.01.2024]

[47] `https://tuwel.tuwien.ac.at/user/managetoken.php?lang=en` [last access: 18.01.2024]

[48] `https://www.tuwien.at/en/tu-wien/organisation/central-divisions/campus-software-development/lehr-und-lerntechnologien-2021/services/tuwel` [last access: 18.01.2024]

As of now, this token is required in each and every API request as the value of „wstoken“-key, which can be seen in Figure 5.2.



```
def mod_assign_get_assignments(self, course_id):
    data_assignments = {
        "wstoken": self.wstoken,
        "wsfunction": 'mod_assign_get_assignments',
        "moodlewsrestformat": 'json',
        "courseids[0]": course_id
    }
    return self.__post_request(data_assignments)
```

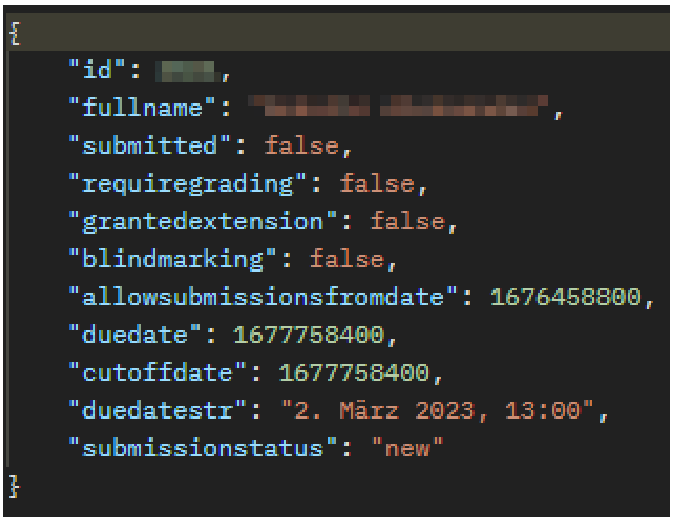
Figure 5.2: API Request example

As a next step, the Postman[49] API Platform (see Section 2.1.3) was used to find useful and informative API calls. Therefore, all provided requests were filtered according to the following criteria:

- The API call must provide some useful information or is working without any errors. Thus, this excludes outputs with empty lists or values.
- The TUWEL admin-role in the affected course called for instance „LVA-Leiter/in“ should be able to invoke the API call without additional permissions by the TUWEL technical support team.
- The API call must provide useful and course-related information, which can be used in a later step.
- The API calls, which provide advanced or additional information will be favored in the filtering process.
- The API call does not edit, assign, delete, or change any configurations or course-related content in the affected course. Additionally, linked events or triggers must not be used, due to unknown effects on the TUWEL course. In other words, only GET-Requests will be part of this research and thus this master thesis to not intervene in any ongoing programming courses.
- No external tools, deprecated API calls, or mobile extensions will be considered in this step.
- Only API calls, which are accessible and used in TUWEL are observed accordingly.

The Figure 5.3 shows the example output when posting an API request.

[49] https://www.postman.com/ [last access: 18.01.2024]



```
{
    "id": [redacted],
    "fullname": [redacted],
    "submitted": false,
    "requiregrading": false,
    "grantedextension": false,
    "blindmarking": false,
    "allowsubmissionsfromdate": 1676458800,
    "duedate": 1677758400,
    "cutoffdate": 1677758400,
    "duedatestr": "2. März 2023, 13:00",
    "submissionstatus": "new"
}
```

Figure 5.3: API Request Output example (sensible data was blurred in this image)

Thereupon, to further preprocess and save the data provided by the API call some elements and values were truncated.

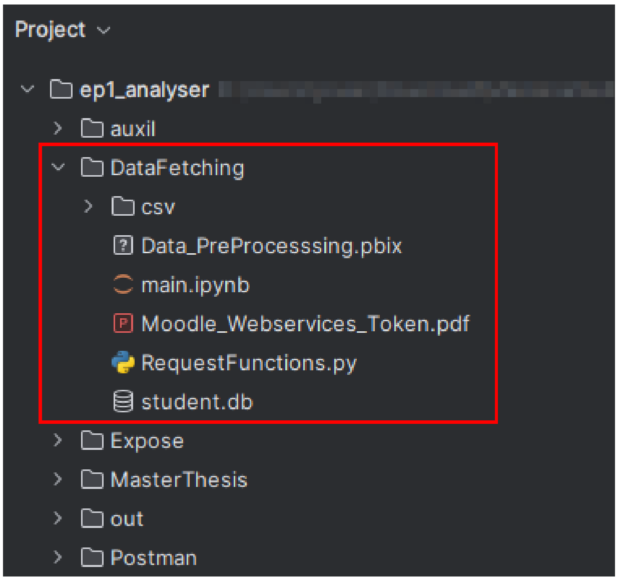
5.1.2 Project and Data Structure

This part describes and focuses on the general structure of the extraction process, which includes information about the input- and output files, which contain student and course information. To ensure usability and reproducibility the Jupyter Notebook-extension (see Section 2.1.3) was used to enhance the flat Python-code by adding informative descriptions and explanations. The overall structure of the Intellij IDEA project can be viewed in Figure 5.4.

Additionally, to increase the readability the helper class „RequestFunctions“ was created, which manages the database connections, cursors, and POST-request to TUWEL's API service. However, the „main“ file contained the code, calculations, and further documentation how the data was structured and saved accordingly. Great value was put on generic development and design, when for instance, if the affected course or semester might be changed, no additional steps must be taken besides changing the desired course name (see Figure 5.5)

In general, the TUWEL API documentation provided a list of attributes, which set the foundation for the latter API requests utilized by the model. This list includes the following attributes:

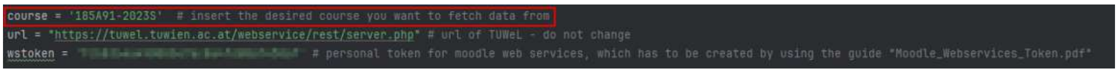
- Required Input-Arguments, optionally with limits (e.g. not negative)



```
Project ⌄

⌄ ep1_analyser [redacted]
  › auxil
  ⌄ DataFetching
    › csv
      Data_PreProcesssing.pbix
      main.ipynb
      Moodle_Webservices_Token.pdf
      RequestFunctions.py
      student.db
  › Expose
  › MasterThesis
  › out
  › Postman
```

Figure 5.4: View of the Project Structure



```
course = '185A91-2023S'  # insert the desired course you want to fetch data from
url = "https://tuwel.tuwien.ac.at/webservice/rest/server.php" # url of TUWeL - do not change
wstoken = [redacted] # personal token for moodle web services, which has to be created by using the guide "Moodle_Webservices_Token.pdf"
```

Figure 5.5: General Data fields in Jupyter Noteboook

- Optional Input-Arguments, optionally with limits (e.g. not negative)
- Output Arguments
- Error messages
- Check for user login
- Callable in AJAX

For the development process the data was maintained and revised using GitLab as mentioned in Section 2.1.2 by pushing changes in an online repository, which was provided by Dr. S. Podlipnig. Different stakeholders can use this platform to check up on the current development process and give feedback if necessary or referring to older versions of the model.

5.1.3 Extraction Process and Data Collection

The following section of this master thesis briefly describes the extraction process on how and in which order the data tables were created. In addition to that, values, parameters, and variables will be explained to get a full view and comprehend the design decisions in some sections of the data collection process.

Before even starting the implementation process some considerations were made to keep the complexity of this model as low as possible and provide long term functionality by using as little external libraries as possible. However, in order to conveniently create requests and manage database connections the „Request HTTP“ library[50] was chosen for this purpose. Furthermore, to manage, add, and edit data frames the popular Data Analysis Library „pandas“[51] and its features were utilized. By combining both libraries, all functions and use cases for the development of this model can be realized, while at the same time providing a fairly simple setup process.

When starting the connection to the database file the first requests were fetched from TUWEL, which are saved in separate data frames. Eventually, a number of columns were dropped, which often contained config-files, attachments, duplicates, and unrelated custom fields. Data or information of basic API calls such as getting the *„course ID“* was used in more specific API calls to fetch student groups or assignments for instance.

Finally, the data was stored in both — a SQL database and in CSV-files to enable interfaces and connections to external services or programs. The database file can be used to view the data tables and identify possible associations within the data. Nevertheless, CSV-files, which are saved in the folder *./ep1_analyzer/DataFetching/csv* as seen in Figure 5.4 will be converted and utilized as data sources for Power BI.

For testing purposes, the IP1 course of the summer term 2023 was uploaded and converted into CSV-files which provide 358MB of student information. This information is divided into 21 different data tables or CSV-files. It is planned to run the developed tool on other semesters of the IP1 course or other courses, which provide sufficient information for further analysis and visual interpretation. Typically, more programming students participate in the winter terms, due to the recommendations in the informatics curriculum[52] of the TU Wien, admission procedures, and exams for Higher School Certificates.

5.2 Transformation and Loading Implementation

The data transformation was conducted in a step-by-step process based on the extracted and imported data in Section 5.1.3. In the first instance, the task was to organize the raw data tables by filtering and renaming data columns to simplify the latter analysis

[50] https://pypi.org/project/requests/ [last access: 18.01.2024]

[51] https://pandas.pydata.org/ [last access: 18.01.2024]

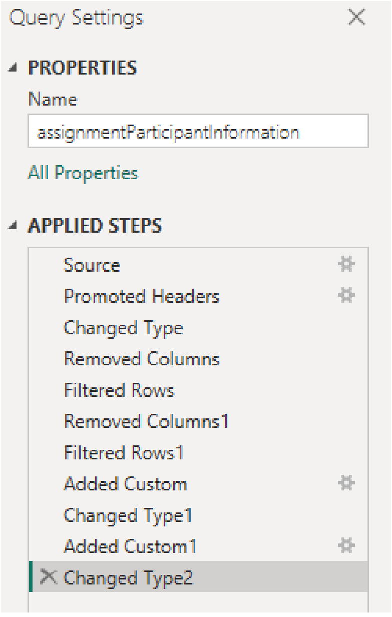
[52] https://tiss.tuwien.ac.at/curriculum/public/curriculum.xhtml?dswid=4011&dsrid=672&key=71647 [last access: 21.01.2024]

and visualization. Power BI offers various transformation processes, which include also more advanced techniques, such as Power BI Data Analysis Expression (DAX) [93] to freely customize and edit data tables. On top of that, new columns including measures, calculations, and column combinations can be used to find differences in datasets and create more advanced visualizations, which are not obtainable otherwise [29]. Thus, for easier reconstruction the following Table 5.1 represents the filtered columns in this process.

Table	deleted columns	Comments
quizUserAttemptReview	state, preview, timemodifiedoffline	
quizUserAttempt	state, preview, timemodifiedoffline	
quiz	intro, introformat, groupmode, groupingid, lang, overduehandling (rule which does not have any impact on the data), preferredbehavior, grademethod, decimalpoints, questiondecimalpoints, shuffleanswers, password, subnet, browsersecurity, hasfeedback, visible	
posts	unread, wordcount, charcount, messageformat, haswordcount, isdeleted, isprivatereply	
participant	grantedextensioin, blindmarking, exception, errorcode, message	
groups	courseid, description, enrolmentkey, idnumber, visibility, participation	
gradeItems		
forum	course, introformat, lang, duedate, cutoffdate, assessed, assesstimestart, assesstimefinish, grade_forum, grade_forum_notify, rsstype, rssarticles, warnafter, trackingtype, blockafter, blockperiod, completiondiscussions, completionreplies, completionposts, cancreatediscussions, lockdiscussionafter, istracked	
discussion	groupid, timestart, timeend, parent, mailed, messageformat, messagetrust, totalscore, mailnow, numunread, pinned, locked, starred, canreply, canlock, canfavorite	
assignmentParticipantInformationRoles		Deleted table
assignmentParticipantInformationGroups	description	
assignmentParticipantInformationEnrolledcourses		
assignmentParticipantInformation	suspended, recordid, grantedextension, description, descriptionformat	
assignment	course, nosubmissions, submissiondrafts, sendnotifications, sendlatenotifications, sendstudentnotifications, completionsubmit, gradingduedate, teamsubmission, requireallteammemberssubmit, teamsubmissiongroupingid, blindmarking, hidegrader, revealidentities, attemptreopenmethod, maxattempts, markingworkflow, markingallocation, requiresubmissionstatement, preventsubmissionnotingroup, intro, introformat, timelimit, submissionattachments	
userSubmission	attemptnumber, timestarted, groupid, latest, status	status only "submitted"
userGradeItem	otucomeid, scaleid, locked, gradehiddenbydate, gradeneedsupdate, gradeishidden, gradeislocked, gradeisoverridden, gradeformatted	
userGrade	maxdepth, courseidnumber, courseid	Deleted table
submissionGradingSummary		Deleted table
studentInformation	course, introformat, groupmode, groupingid, lang, tobemigrated, legacyfiles, legacyfileslast, displayoptions, filterfiles, contentfile_filepath, contentfile_isexternalfile	
quizUserQuestions	timemodifiedoffline, slot, type, page, questionnumber, number, html, sequencecheck, lastactiontime, hasautosavedstep, flagged, status, blockedbyprevious, mark, maxmark, settings, attemptid, preview	

Table 5.1: Summary of filtered columns using Power BI during the data transformation process

As mentioned above, one of the main objectives focused on adaptability and usability when changes need to be taken by one of the stakeholders to get further insights into the course and student data. For this purpose, additional data cleansing and type changes were implemented and custom columns were added using Power BI to use this information in the upcoming visualization steps. As shown in Figure 5.6 Power BI offers a feature to list the changes chronologically, which increases the transparency.


Query Settings

PROPERTIES

Name: assignmentParticipantInformation

All Properties

APPLIED STEPS

- Source
- Promoted Headers
- Changed Type
- Removed Columns
- Filtered Rows
- Removed Columns1
- Filtered Rows1
- Added Custom
- Changed Type1
- Added Custom1
- Changed Type2

Figure 5.6: Power BI Applied Steps list

Moreover, this master thesis and its linked model intends to support the teaching process and give valuable insights of students, which are at risk of dropping out. Therefore, no pseudonymization of student or course data will be conducted to identify the struggling students. This approach was discussed with Dr. S. Podlipnig and therefore access would only be granted for certain course staff to ensure data safety and avoid malicious activities. However, for this master thesis any personal information or hints to track down certain students will be blurred out or made indecipherable effectively.

Lastly, storing the data must be considered to ensure a high level of usability and also compatibility with Power BI's import capabilities and provided external services. In this case, Power BI automatically provided integrations in all three — the Report-, Table-, and Model View. Changes in either the import process or underlying data were automatically replicated in these three views.

5.2.1 Implementation Details in PowerBI

In the current state, most of the uploaded tables had no relations to other data, which decreased the possible capabilities and visualizations in respect to usability even for domain experts. Thus, for this step Power BI's Model View was utilized to arrange and connect single tables. Generally speaking, the Model View was used for documentation for other stakeholders and for development or debugging of student data. The main goal is to create a legible overview, which does not require all possible relations to other tables, which can be viewed in Figure 5.7.

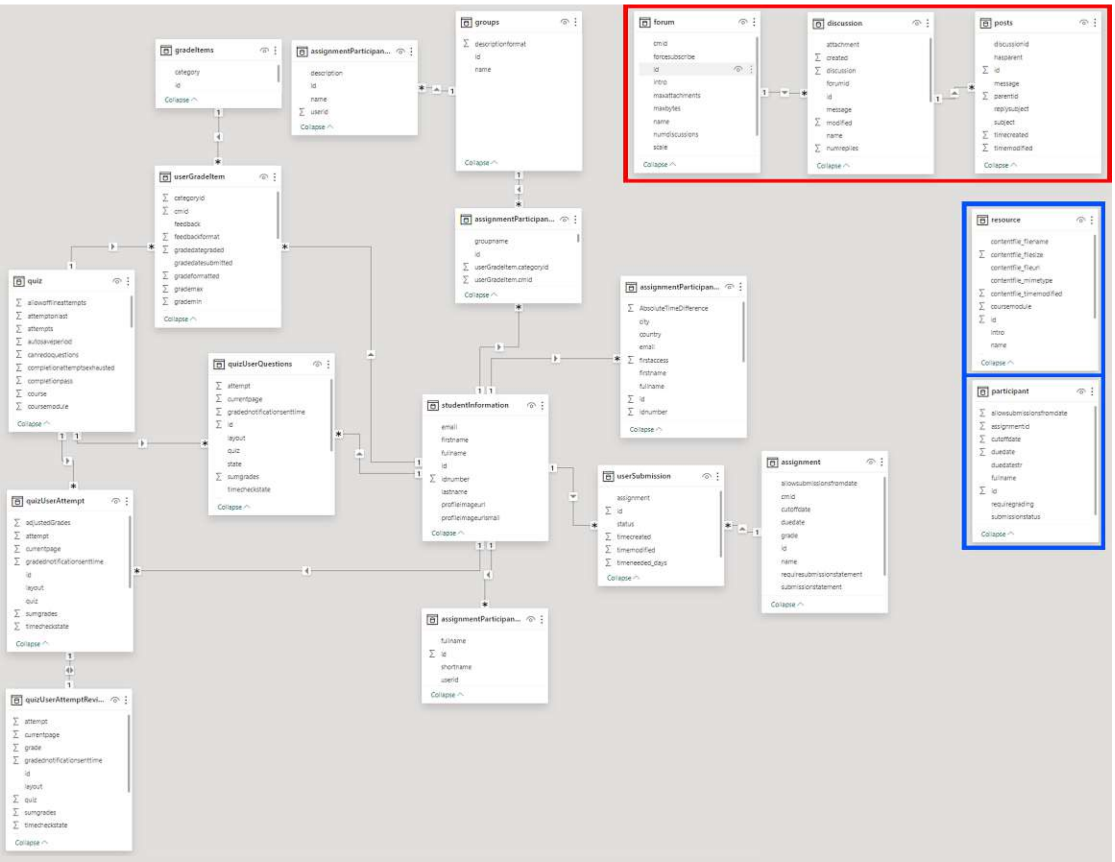


Figure 5.7: Power BI Model View

As seen in the figure, relations to *participant* or *resource* are not providing sufficient or valuable information and can simply be used for labels or headings separately. This section was highlighted in a blue box (see Figure 5.7) — the red box represents the relations to the forums, which consists of discussions and any TUWEL participants with adequate access permissions are able to create posts in discussions or even open up new discussion-sections. These fields exhibit intersections with for instance student information, but these connections are not relevant for this master thesis and also practical use. However, Power BI's GUI and its features offers a very dynamic and adaptive design, which comes in handy for university staff such as tutors and lecturers.

The Power BI Table View can be used to gain more insights in student and course data by filtering certain properties. These properties can be valuable when creating visualizations and trying to find initial patterns in datasets.

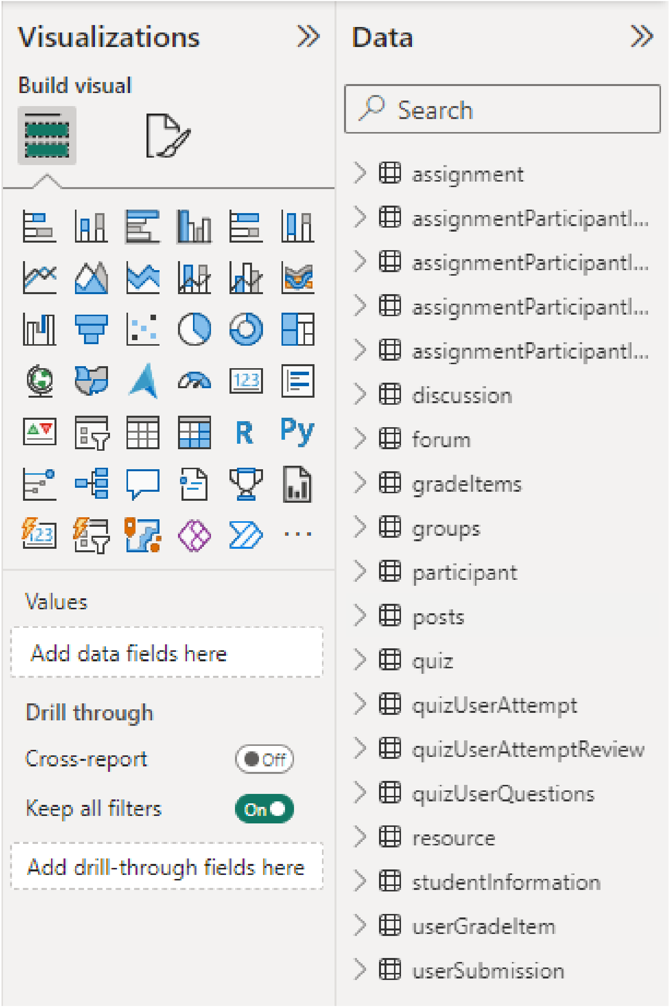
5.3 Data Loading and Visualization

This section represents the last step of the ETL process as described in Section 5, which now loads sample data in Power BI. The first step would be to get datasets by using one of the supported data-sources, in this case CSV-files, which are automatically converted into tables in Power BI. At this point, the import process can be changed and integrations to well-established databases are also supported, which allows a high level of customization.

As discussed in Section 5.2 typically some pre-processing steps need to be taken to simplify latter analysis and visual interpretation. Depending on which programming course is chosen for this step, small adaptations must be made in the Model View and in the data pre-processing step. An example would be if the stakeholder wants to focus more on student contribution in forums or share in discussions. A big advantage of Power BI is the already mentioned „Applied Steps list“ in Section 5.2, which enables a convenient traceability for similar or previous courses. In this instance, loading data from new or previous semesters of a course might need only little to no adaptations to visualize the data properly, if the overall structure of the course has not changed significantly.

Subsequently, Power BI Desktop offers a combination of these three process steps — Data Exploration and Transformation, Data Loading, and Data Visualization. The visualization can be done in the Report View, which provides by default a range of different visualizations that can be combined with the previously loaded data. As mentioned in Section 3.2.2 these visualizations could be used and combined with the pre-processed data tables to create an informative report or dashboard for presentations or to gain insights in the data to implement an informed decision-making process (see Figure 5.8). This section of this master thesis shows a number of visual examples by using the IP1 course in the summer term of 2023 in Section 6 — however, critical data or student information will be blurred or made indecipherable in another way.

Lastly, to increase the level of interaction and provide a clear view, support elements such as *treemaps* (Figure 5.9), *cards* (Figure 5.10), *Key Influencers* (Figure 5.11), etc. were added to the visual representations. Special elements such as Treemaps enable viewers to



Cross-report Off

Keep all filters On

Figure 5.8: Power BI Report View Visualization Options

filter the data according to pre-specified categories and get a closer look on smaller data samples for instance to analyze different student-groups.

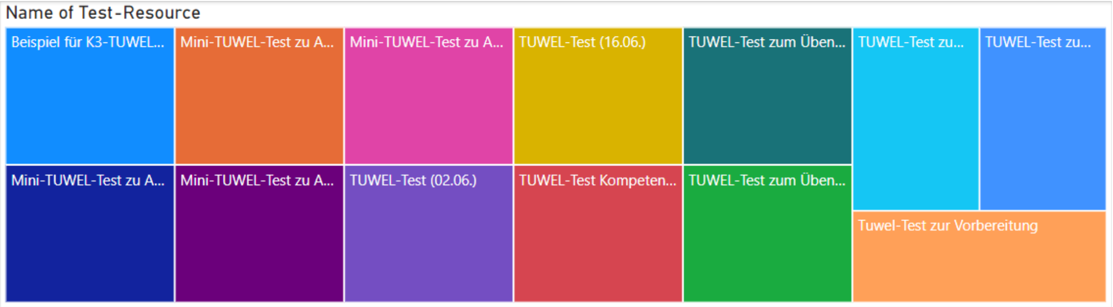


Figure 5.9: Power BI Treemap example

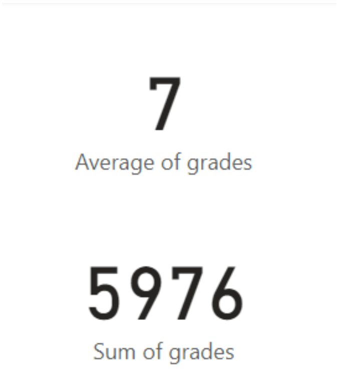
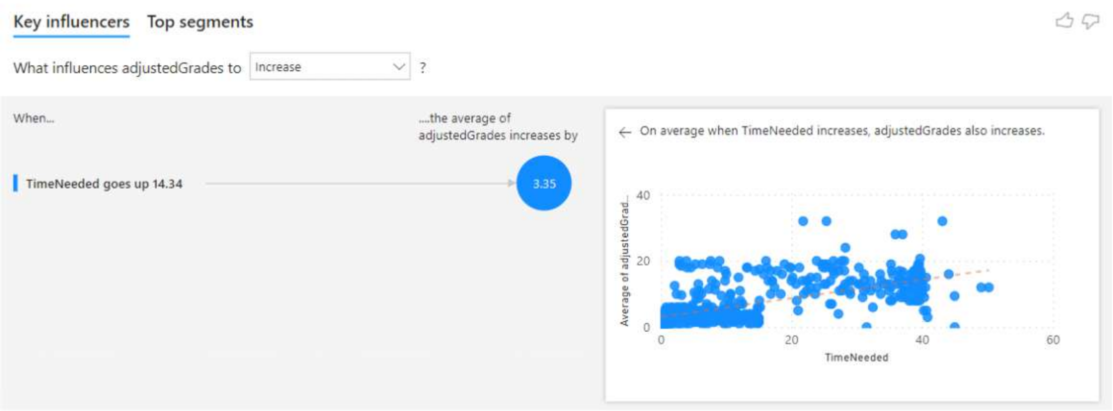


Figure 5.10: Power BI Card examples



What influences adjustedGrades to Increase

...the average of adjustedGrades increases by

TimeNeeded goes up 14.34

On average when TimeNeeded increases, adjustedGrades also increases.

Figure 5.11: Power BI Key Influencers examples

CHAPTER 6

Education Intelligence Visualization

Based on the ETL processes and data in the previous Section 5, this chapter provides a closer look on the visualization concepts and sharing of analysis results. Similar to the approach by Z. Liu et al. [94] the processed data will be adapted to specific problems or needs, thus not all data tables might be used in the visualization process. The ultimate goal is to improve the efficiency of visualizing student or course data and developing an interactive environment for lecturers, tutors, and instructors. The foundation and initial information is based on H. Baars and H. Kemper's book [95] on BI and analytics.

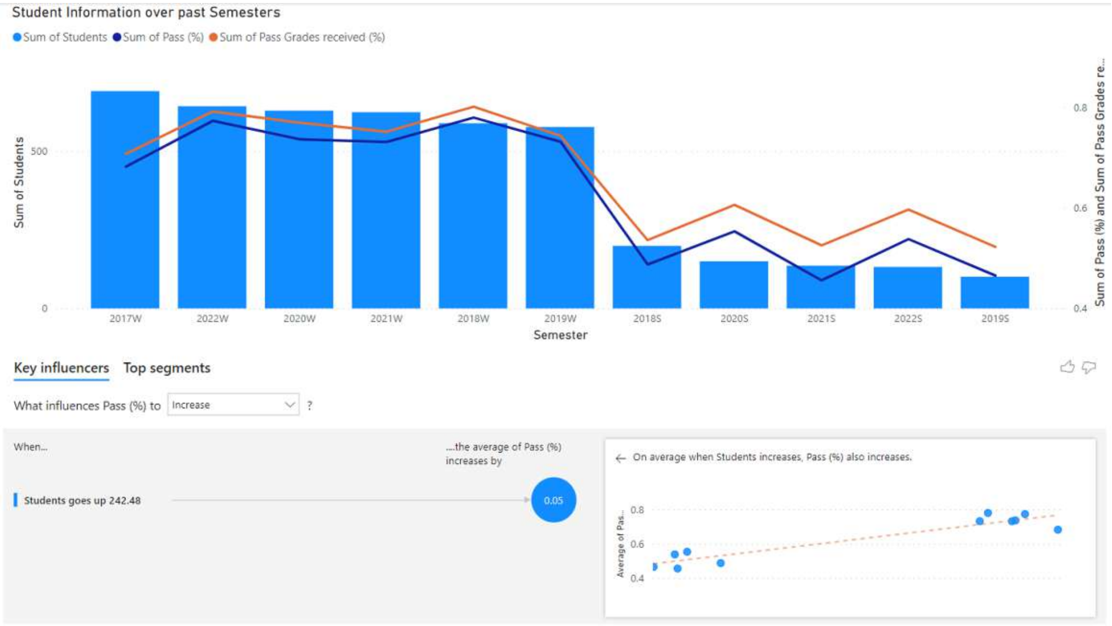
The following section provides a brief insight in the characteristics and core aspects of the IP1 course, which will be used as a case study of the developed indication model, including a visual representation.

6.1 Data Visualization of Introduction to Programming 1

As mentioned before, this section of this master thesis will provide overall information and statistics of the IP1 course — the summer term of 2023 of this course will be used to test the developed visualization tool and statistical model and to answer the research questions and compare the results in Section 7. Thus, this part shows a high-level overview of the course to gain further insights into absolute student and pass numbers, which is displayed in Table 6.1.

Semester	Students	Pass (%)	K1 pass	K1 all	K1 pass (%)	K2 pass	K2 all	K2 pass (%)	K3 pass	Grades received	Pass Grades received (%)
2017W	691	68,16%	317	515	61,55%	154	176	87,50%		666	70,72%
2018S	199	48,74%	76	172	44,19%	21	27	77,78%		181	53,59%
2018W	589	77,93%	378	502	75,30%	81	87	93,10%		573	80,10%
2019S	101	46,53%	37	90	41,11%	10	11	90,91%		90	52,22%
2019W	577	73,14%	317	456	69,52%	105	121	86,78%		568	74,30%
2020S	150	55,33%	71	134	52,99%	12	16	75,00%		137	60,58%
2020W	629	73,61%	346	503	68,79%	117	126	92,86%		602	76,91%
2021S	136	45,59%	48	120	40,00%	14	16	87,50%		118	52,54%
2021W	624	73,08%	315	476	66,18%	85	92	92,39%	56	607	75,12%
2022S	132	53,79%	63	124	50,81%	2	2	100,00%	6	119	59,66%
2022W	643	77,29%	332	474	70,04%	84	88	95,45%	81	628	79,14%

Table 6.1: List of student pass numbers of the Introduction to Programming 1 course


Student Information over past Semesters
● Sum of Students ● Sum of Pass (%) ● Sum of Pass Grades received (%)
Sum of Students
500
0
2017W
2022W
2020W
2021W
2018W
2019W
2018S
2020S
2021S
2022S
2019S
Semester
0.8
0.6
0.4
Sum of Pass (%) and Sum of Pass Grades re...
Key influencers
Top segments
Average of Pas...
0.8
0.6
0.4

What influences Pass (%) to Increase ?

When...

...the average of Pass (%) increases by

Students goes up 242.48 → 0.05

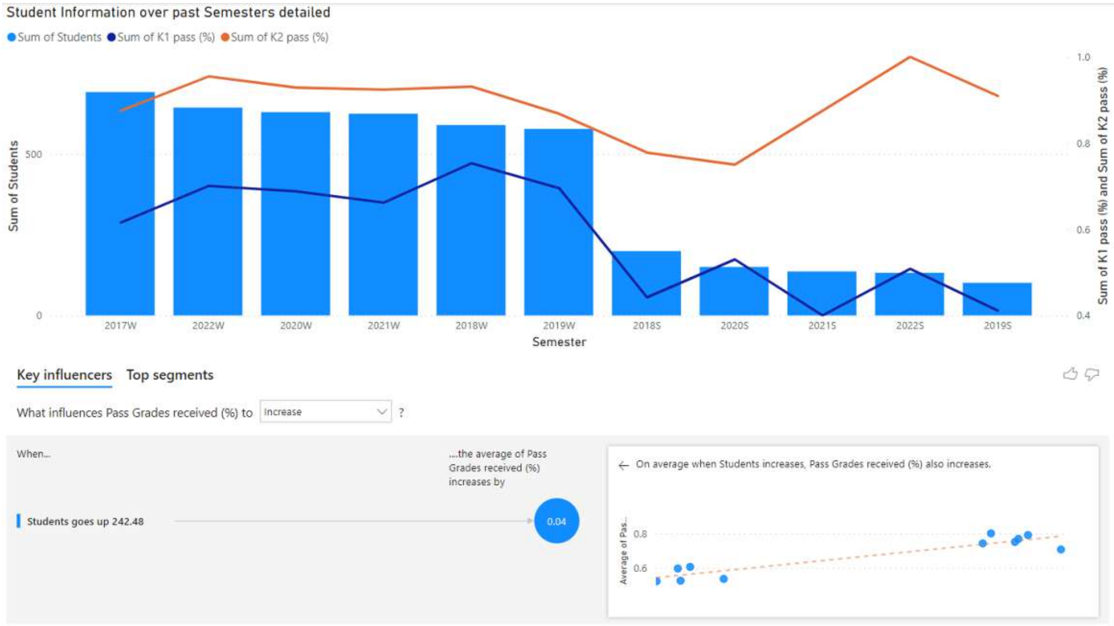
← On average when Students increases, Pass (%) also increases.

Figure 6.1: Student information by semesters of the IP1 course

The Table 6.1 was provided by Dr. S. Podlipnig, who collected general course statistics, starting from the winter term of 2017. In Figure 6.1 a dashboard was created, which displays the general student distribution and pass rates. In general, Figure 6.1 and upcoming figures are exemplary representations, which primarily showcase the capabilities of the developed visualization tool and latter statistical model. Furthermore, this master thesis gives an insight into the possible visualization techniques, which in some cases could be replaced with more suitable representations, when for example considering expert knowledge of the affected course. The course IP1 is offered in every semester — thus, semesters ending with „W“ stand for winter terms and the summer terms end with „S“. However one can observe, that more students participate in winter terms, which is due to the admission procedure and exams for Higher School Certificates, which are required to begin any bachelor studies at the TU Wien[53].

Additionally, both lines — the blue and orange line show the pass-rate of students, which is about 23.87 percentage points higher in winter term courses. Below the first graph, Power BI offers a pre-built visualization to determine the influences on student grades. In this view, students, who participate in winter terms are more likely to pass the course, than students participating in the summer term. However, some students do not receive any grades in the IP1 course, because they did not submit any assignments. This difference can be seen in the mentioned figure by comparing the second y-axis, that represents the relative pass rates. Similarly, the summer term exhibits a higher relative

[53] https://www.tuwien.at/studium/zulassung [last access: 21.01.2024]


Student Information over past Semesters detailed
● Sum of Students ● Sum of K1 pass (%) ● Sum of K2 pass (%)
Sum of Students
500
0
1.0
0.8
0.6
0.4
Sum of K1 pass (%) and Sum of K2 pass (%)
2017W
2022W
2020W
2021W
2018W
2019W
2018S
2020S
2021S
2022S
2019S
Semester
Key influencers
Top segments
Average of Pas...
0.8
0.6

What influences Pass Grades received (%) to Increase ?

When...

...the average of Pass Grades received (%) increases by

Students goes up 242.48

0.04

← On average when Students increases, Pass Grades received (%) also increases.

Figure 6.2: Student information by semesters of the IP1 course (detailed)

number of students, who do not submit any assignments, thus receive no grade in this course.

Next, Figure 6.2 displays a more detailed view on the pass rates regarding K1 and K2 groups. Interestingly, no significant patterns can be assessed in K2-groups, which is colored in orange in this comparison. In contrast, it can be observed that K1-groups exhibit higher pass rates in winter terms of the IP1 course. Now by comparing both pass rates, it can be seen that K2-groups, which often have programming experience, are on average 30.80 percentage points better in the introductory programming course.

As described in Section 2.2.3 the IP1 course usually offers nine assignments or exercises, excluding students in K2- and K3-groups, which can skip the first or all exercises accordingly. Depending on the assignment, between two and seven points can be reached, which is also a crucial step for achieving a positive grade (more information will be provided Section 7). The Table 6.2 displays a summary of these aspects divided by „Ticked Exercises“, „Attendance“, and „Points for Presentation“.

Each „Aufgabenblatt“ and „Üben“ (see Table 6.2) is mapped to several programming concepts or topics, pre-defined and slightly adjusted in the respective semester of the IP1 course. For this purpose, every assignment is based on previous lectures and contains a section describing the targeted programming concepts. The following list displays the

Assignment/Exercise	Reachable Points	Ticked Exercises	Attendance	Points for Presentation
Üben 1	2	2		
Üben 2	2	2		
Üben 3	2	2		
Aufgabenblatt 1	6	4,4	82%	98,24%
Aufgabenblatt 2	6	4,12	78%	93,50%
Aufgabenblatt 3	6	3,95	72%	96,71%
Aufgabenblatt 4	6	4,1	73%	97,81%
Aufgabenblatt 5	6	3,4	64%	92,00%
Aufgabenblatt 6	7	4,2	63%	94,42%

Table 6.2: Descriptive Analysis of Exercises and Assignments

mapping for the mentioned term of the IP1 course — assignments are denoted as „A“ and exercises are denoted as „E“ as shown below:

E1 : Variables, data types, operations, conditions

E2 : Conditions, loops

E3 : Input and output, CodeDraw, debugger, loops

A1 : Conditions, simple loops, external classes String and CodeDraw

A2 : Loops and nested loops, drawing with CodeDraw, implementation of methods, Scanner-class

A3 : Code analysis and implementation-style, implementation of methods, overloading of methods, recursion, recursion with CodeDraw, comparison of recursion and iterative implementation

A4 : Code-understanding with arrays, one- and two-dimensional arrays, recursion with one-dimensional arrays

A5 : One- and two-dimensional arrays, recursion, graphic representation, two-dimensional pictures

A6 : One- and two-dimensional arrays, methods, graphic representation, game logic

In the summer term of 2023, a decrease on the average attendance on latter assignments can be distinguished, which results in less points on average by the end of the course. However, it is not possible via TUWEL to split the programming concepts completely, because only the sum of points for each assignment is inputted in the mentioned management platform by university staff. For further reference, the Table 6.3 shows the distribution of points of all offered assignments.

Additionally, two so-called „Mini-TUWEL-Test“ are added to the *Aufgabenblatt 1* and *Aufgabenblatt 2*.

Assignment/Exercise	Points	Crosses x Attendance x Presentation	Crosses x Presentation
Aufgabenblatt 1	4,4	3,54	4,32
Aufgabenblatt 2	4,12	3,00	3,85
Aufgabenblatt 3	3,95	2,75	3,82
Aufgabenblatt 4	4,1	2,93	4,01
Aufgabenblatt 5	3,4	2,00	3,13
Aufgabenblatt 6	4,2	2,50	3,97

Table 6.3: Summary of Assignments scaled by Attendance and Presentation

Ultimately, these exercises and „Mini-TUWEL-Test“ constitute 50% of the the final grade.

The other 50% of the IP1 course can be achieved by completing the TUWEL-exam and programming exam, which is offered twice for every student. In more detail, students in the summer term of 2023 got 17.03 points on average on the first term and 11.16 points on average on the second term. The maximum number of achievable points amounts to 30 points per test and the better results counts for the final score of the IP1 course. Similar to this, students performed worse on the second TUWEL-Test with only 12.69 points on average and 13.55 points on average on the first term — the maximum number of achievable points amounts to 20 points per test and analogously the better result counts.

CHAPTER 7

Evaluation and Results

The following chapter of this master thesis gives a detailed insight into the results of the case study on the IP1 course, which was used to test the developed tool to indicate student performance within a semester. The first Section 7.1 outlines overall information about the used hardware- and software landscape. The Section 7.2 presents the analysis results and visualizations, which can be interpreted as an estimation and first assessment of the tool's performance. Finally, the last part of this chapter (Section 7.3) is about the evaluation of the developed model, which provides additional information and an assessment by several domain experts.

7.1 Execution Landscape

The extraction process and initial pre-processing was executed on the following specifications:

- Central Processing Unit (CPU): 11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz
- Random-Access Memory (RAM): 16GB 2400MHz DDR4
- Memory Storage: 500GB M.2 SSD
- Operating System (OS): Windows 10 Pro 64-Bit

The developed program was implemented and configured using the Python interpreter version 3.10 as described in Section 5.1.3. The Intellij IDEA (see Section 2.1.1) provided useful integrations, which were used for the creation of the execution environment — ultimately „venv“[54] and its features were utilized in this instance. Generally speaking,

[54] https://docs.python.org/3.10/library/venv.html [last access: 25.01.2024]

the library „venv“ enables the development on lightweight virtual environments and manages the usage and integration of additional Python packages.

The goal of this application is to indicate student’s programming performance within semesters, thus the provided data was limited to single semesters and the available information of the affected TUWEL course. The more information was provided and tracked in TUWEL, the more insights can be achieved using jupyter notebook.

As of now, the tool fetches data by using TUWEL’s Webservices, which takes some time to execute — in order to get a rough speed-benchmark sample data from the winter term 2021 was used, which comprises 624 students as described in Section 6.1. Using Jupyter Notebook for the execution of the main Python file, each and every code-block was executed separately, that can be summed up to 1 hour 52 minutes and 54 seconds of net execution time. The high execution time is attributable to the connection and download speed using the API and the underlying code to retrieve the data. By considering the circumstances, currently no speed-up or parallelization processes [96] are required — this model will not be executed several times within a single day according to Dr. S. Podlipnig.

7.2 Data Exploration and Analysis

As depicted in the very first parts of this master thesis, such as Section 1.1 the following section provides a detailed understanding of the capabilities and also limits of this early indication and analysis tool. These examples are aligned with the interests of the main stakeholders of the conducted case study on the IP1 course. Additionally, the latter insights represent the practical part of this work, which might be a useful foundation for the EDM-community and future work, when considering LMS systems such as the Management Platform Moodle. However, due to the generic development, also other courses can be analyzed using this method, without requiring extensive labor or time resources.

A combination of the two announced approaches in Sections 4.2 and 4.3 will be utilized to answer the research questions (see Section 1.1) of this master thesis.

7.2.1 Grade Analysis

The first and most essential part of this analysis and latter evaluation is the grade analysis using Power BI. The collected information and insights will be mainly utilized to answer the mentioned research questions, which represents the practical bit of this master thesis using the IP1 course. Also, this case study represents the capabilities of the developed tool for visualizing student data and determining the student’s performance, which can be compared with other semesters in future work.

Grade (Austria)	Grade (Code)	Verbal	Definition
1	S1	Excellent	if g $\geq$ 92 points
2	G2	Good	if 83 $\leq$ g $\leq$ 91
3	B3	Satisfactory	if 74 $\leq$ g $\leq$ 82
4	G4	Pass	if 65 $\leq$ g $\leq$ 73
5	N5	Fail	if g $<$ 65

Table 7.1: Grading scheme of the Introduction to Programming 1 course

Most of the following visualization base on the initial grading scheme of the IP1 course, which can be viewed in Table 7.1 for the summer term of 2023. As seen in Figures 7.1 and 7.2 student grades can be viewed in one chart — to showcase an example, the maximum number of achievable points for the programming exam is 30, which aligns with distribution of points in Section 2.2.3. Besides the chart the average of grades is displayed as 17.03, thus on average participating students received slightly more than 17 points, which can be converted into 56,67% of the programming exam.

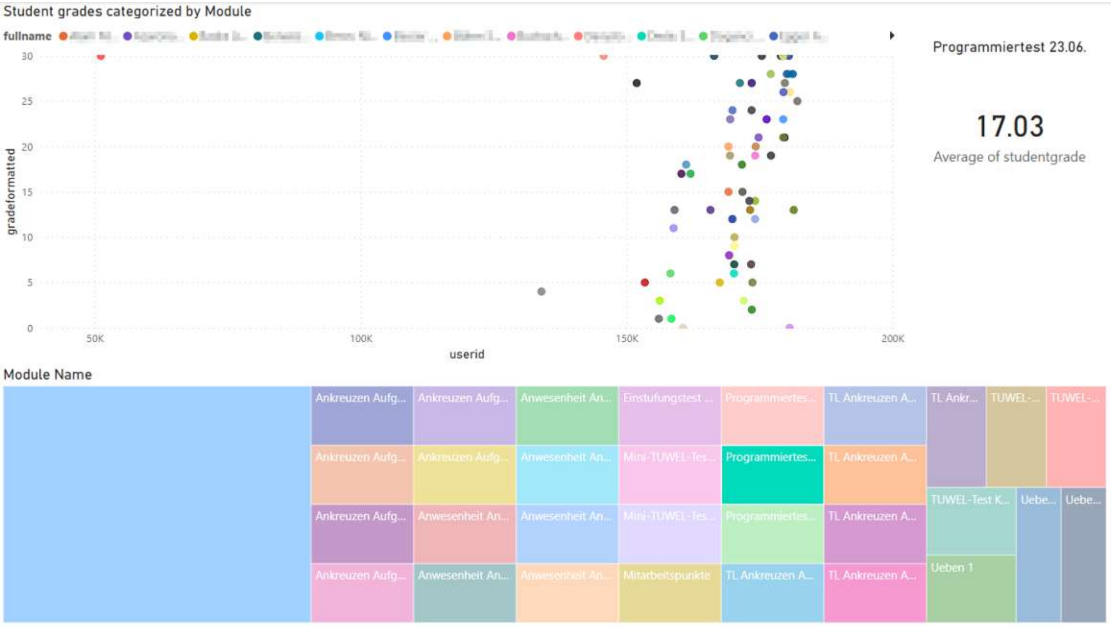


Figure 7.1: Student Grade by Module example 1

In comparison, Figure 7.2 shows the results of the second practical exam, which took place a few days after the first exam. Firstly, less students participated on the second exam and secondly the average grade decreased as well. In other words, students in the second exam term performed on average worse than in the first exam term. However, a small number of students received the majority of points — these observations can be seen as visual outliers.

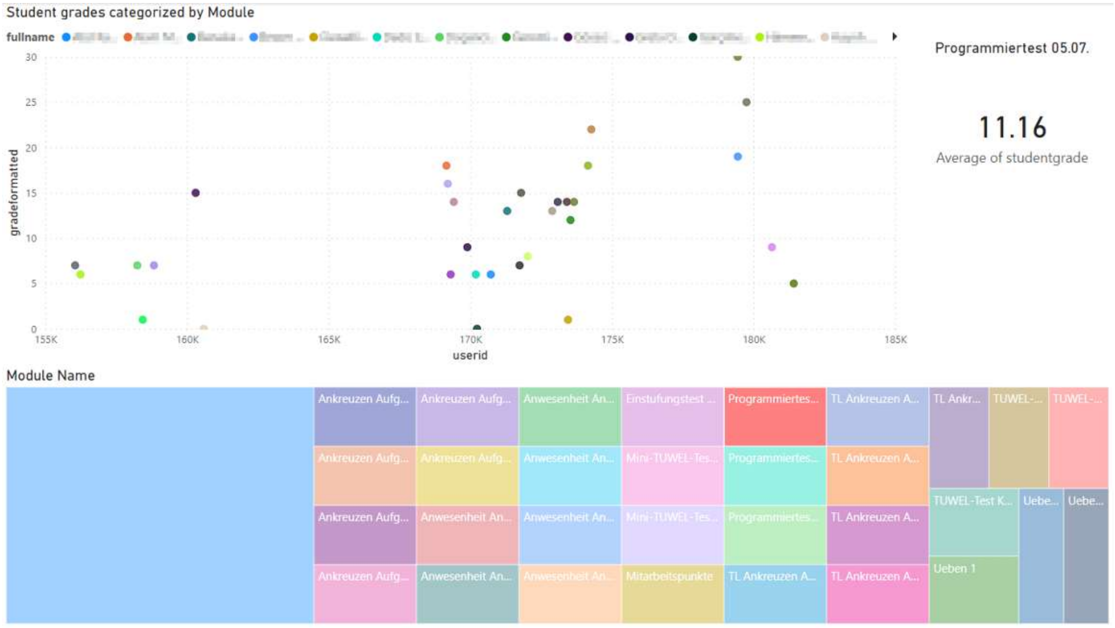
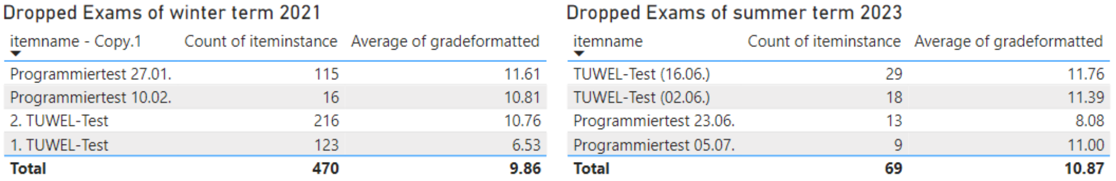


Figure 7.2: Student Grade according to Module example 2

Due to the Structure of the IP1 course (see Section 2.2.3) students were able to participate on both exams and the best result was used for grading. Hence, exams where students received less points were dropped — in addition to the previous charts the Figure 7.3 shows the count of drops per test for both, the winter term 2021 and the summer term 2023. Interestingly, the findings vary between TUWEL online-quizzes and programming exams.

In online-quizzes less students improved their quiz grade in the second exam. In more detail, students achieved 75.61% less points in the summer term of 2021 and 61.11% less points in the winter term of 2023. Conversely in programming exams, students performed better and achieved on average more points on the second exam, which means less drops in programming exams on second terms - concretely 86.09% less drops for the winter term of 2021 and 30.77% on the summer term of 2023.

As a next step, the average student grades were analyzed by the student's total matriculation time in Figure 7.4. Similar to Section 2.3.1 the chart exhibits a dual y-axis — grades of assignments or exams are displayed on the left side as bar charts and the matriculation time is represented as the line chart on the right y-axis. Most of the students with a low matriculation time receive on average more points, while some students with very high matriculation times did not submit any assignments or took part on any exams during



Dropped Exams of winter term 2021

itemname - Copy.1	Count of iteminstance	Average of gradeformatted
Programmiertest 27.01.	115	11.61
Programmiertest 10.02.	16	10.81
2. TUWEL-Test	216	10.76
1. TUWEL-Test	123	6.53
Total	**470**	**9.86**

Dropped Exams of summer term 2023

itemname	Count of iteminstance	Average of gradeformatted
TUWEL-Test (16.06.)	29	11.76
TUWEL-Test (02.06.)	18	11.39
Programmiertest 23.06.	13	8.08
Programmiertest 05.07.	9	11.00
Total	**69**	**10.87**

Figure 7.3: Dropped Exams in winter term 2021 and summer term 2023

the semester. The Treemap (see Section 5.3) of the Figure 7.4 filters data in the plots on the left hand side, which compares the received grade of the specified course with the matriculation time of students in the IP1 course. As described in Section 6.1 students do not receive any negative grade and can register for the IP1 course without penalties.

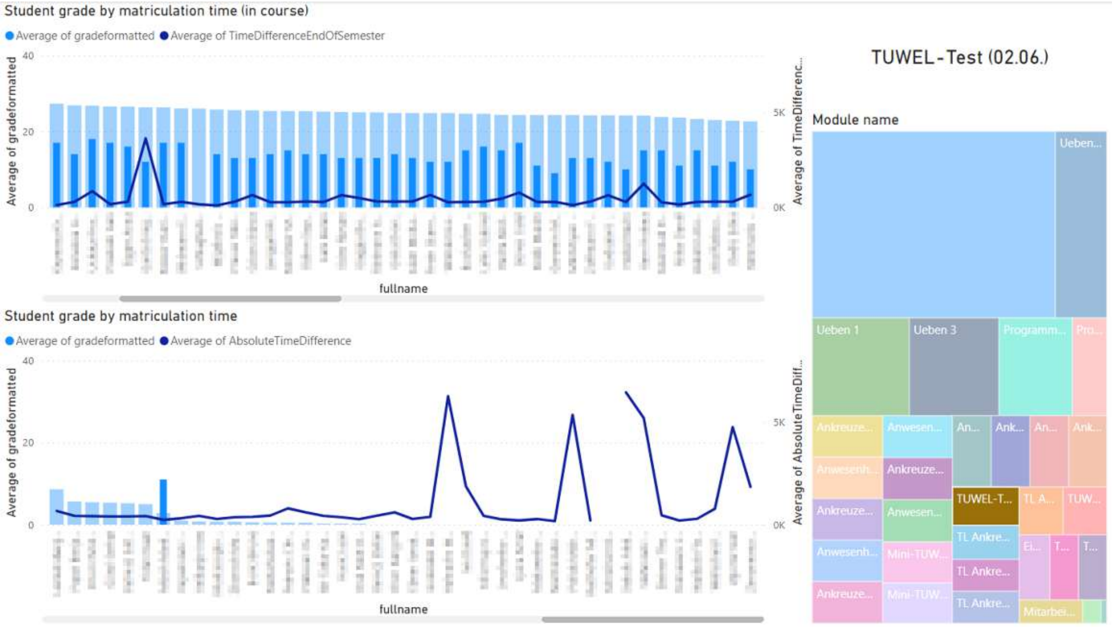


Figure 7.4: Student Grade according to Time

Figure 7.5 displays the average grade over all available quizzes in the IP1 course, which were conducted in TUWEL using the Quiz-Module. Generally speaking, TUWEL already provides various features to inspect and analyze quiz-attempts by students — such a detailed analysis is not possible using the developed tool due to pre-defined permission. These permissions requires the lecturer, who is using the developed tool, to be logged in in each and every student account to inspect the student's quiz-attempt. However, the view can be edited or circumscribed by clicking on the desired TUWEL-quiz, which then displays the average grade of the online exam.

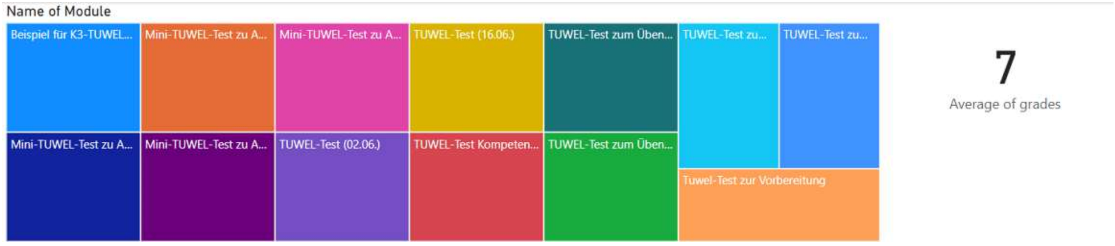


Figure 7.5: Student Grade Averages of TUWEL Quizzes

A more detailed view on TUWEL exams and possibly programming exams can be viewed in Figures 7.6 and 7.7, which compare the student exam grades with the time required (in minutes) to submit the selected exams. Similar to Figure 7.4 the chart exhibits a dual y-axis — in these examples the time was visualized using a bar chart for every student and grades of the TUWEL exam were represented by the line chart. This visualization shows that some participants did not participate in the online exam, which was not graded. Thus, gaps in the line chart can be noticed in these cases. Additionally, the average grade can be seen for the selected exam, which amounts to 13.55 points.

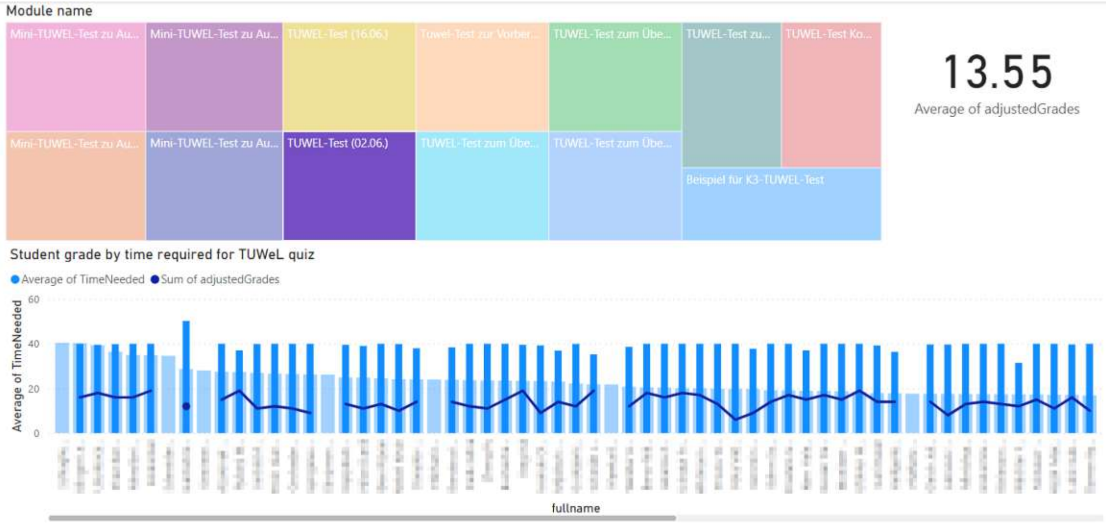


Figure 7.6: Student Grade in TUWEL Quizzes according to Time example 1

7.2.2 Forum Activities

Furthermore, the TUWEL API documentation enables to view and analyze forums, which contain discussions and posts. However, for the purpose of this master thesis only a high-level view is required, as seen in Figure 7.8. In the case of the IP1 course, only

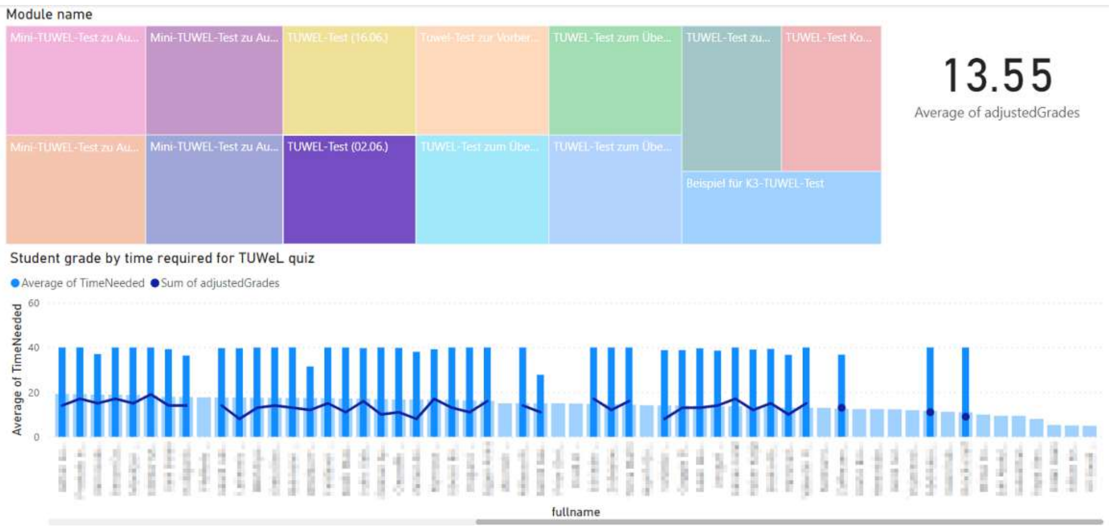


Figure 7.7: Student Grade in TUWEL Quizzes according to Time example 2

two forums were initially created by course admins — the so called „Nachrichtenforum“ is reserved for important messages, alerts, or reminders and only student-staff can start discussions in the mentioned forum. According to the chart, the discussions in the „Diskussionsforum“ can be initialized by all course members and analogously all course members can reply to these discussions as posts.

The number of posts or replies on specific discussions can be observed by considering the line chart. As mentioned in Section 3.2.1 it is also possible to create discussions anonymously by using an implemented extension integrated by the TUWEL development team. This extension enables the use of an anonymous forum, in which students are able to create and reply to discussions anonymously. Nevertheless, in the current version of TUWEL it is not possible to retrieve data from these forums. Integrating this feature might be an useful addition for courses at the TU Wien.

7.2.3 Assignment Analysis

Beside forum activities in Section 7.2.2 and Grade Analysis in Section 7.2.1 the required time to submit assignments can be captured using the provided data in the model. For this visualization the following two rules were applied to avoid data shifts:

1. Submissions which were not submitted shall not be included.

2. Submissions which were submitted in less than 0.01 days shall not be included.

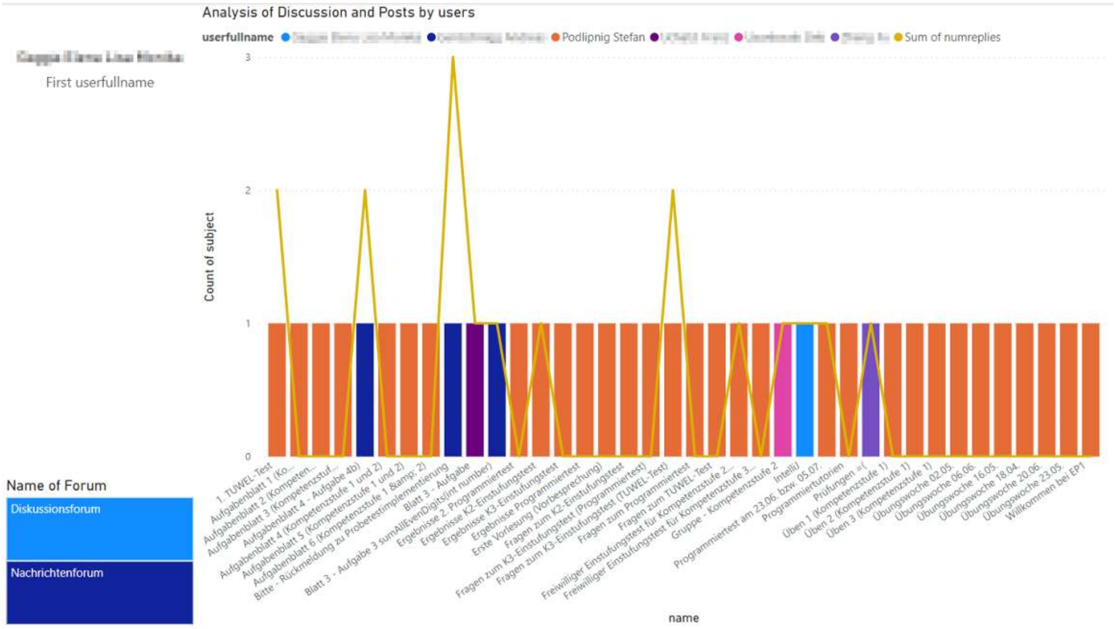


Figure 7.8: Analysis of Discussions and Posts categorized by Students

Assignment number	Average days to submit	Median
Aufgabenblatt 1	8.01	6
Aufgabenblatt 2	5.05	5.13
Aufgabenblatt 3	4.63	4.84
Aufgabenblatt 4	4.09	2.37
Aufgabenblatt 5	5.83	7.07
Aufgabenblatt 6	5.09	2.81
Overall Average	**5.45**	**4.7**

Table 7.2: Descriptive Analysis of assignment submission time

The result of average submission times can be viewed in Table 7.2, which includes a median. This table reveals that „Aufgabenblatt 1" has a slightly higher average time and median value. Reasons for this outlier can be attributed to the prolonged submission time for students or getting used to this type of programming tasks — this might be a good starting point for further research or work.

The Figure 7.9 gives further insights into the submission of „Aufgabenblatt 1", which also shows the distribution of points. Additionally, the required time and standard deviation in days is adduced in this plot, which mainly focuses on submission times for assignments or bigger projects. The required time in minutes can be useful for programming exams or short tests to determine if time was an overall issue during TUWEL-tests for instance.

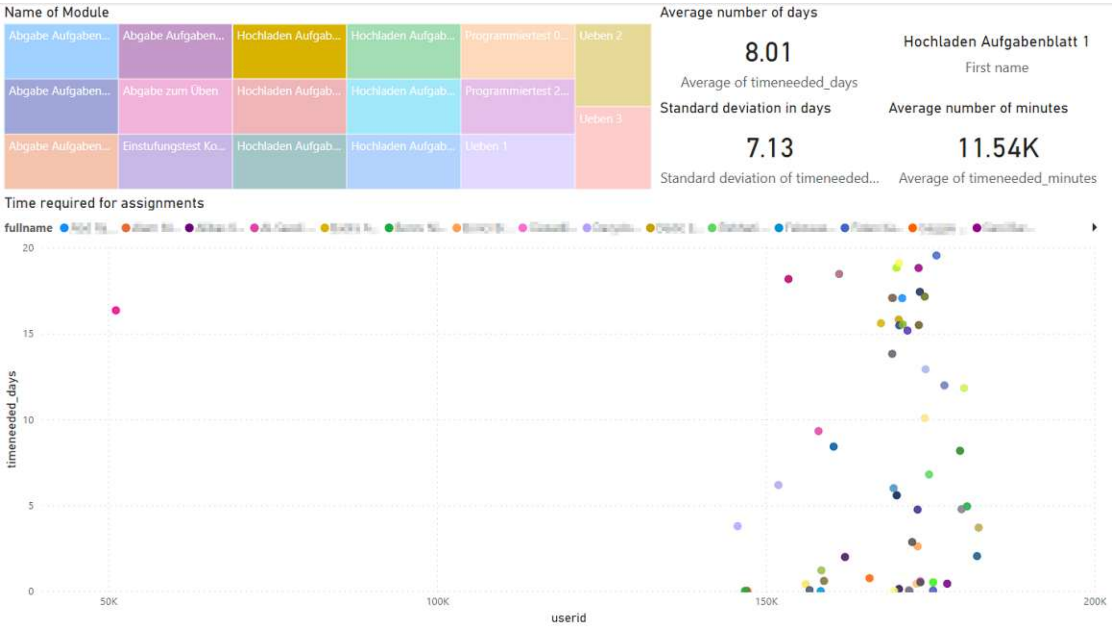


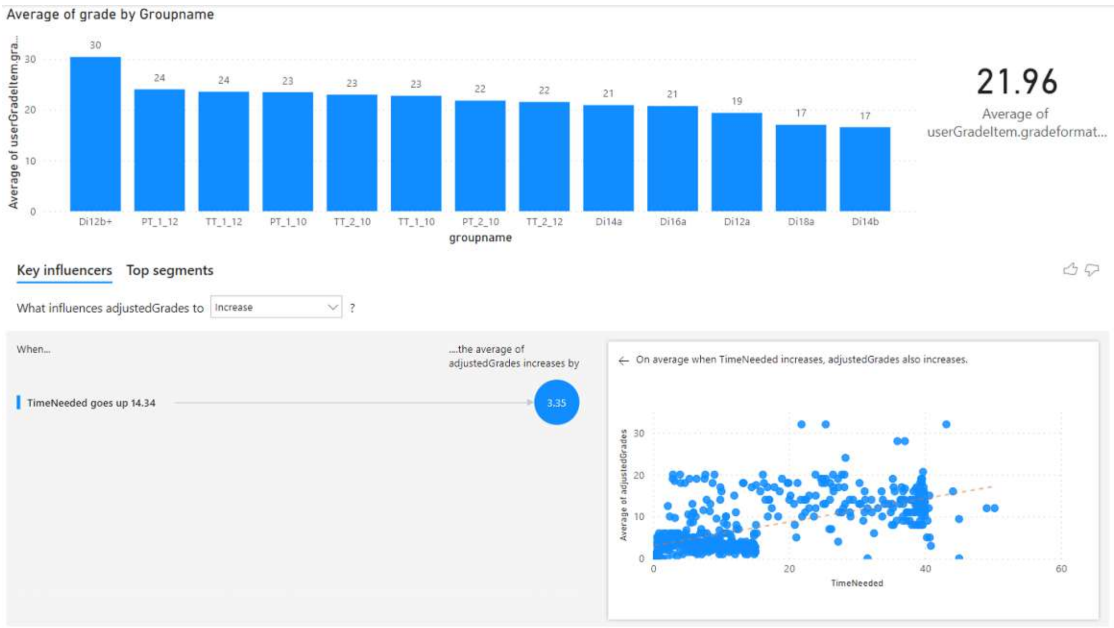
Figure 7.9: Time required for assignments example

7.2.4 Distribution of TUWEL Groups

Lastly, Figure 7.10 shows the student performance according to specific groups. In this chart one can notice, that „Di12b+“, which is a K2-group did significantly better than other groups such as „Di14a“, „Di12a“ — these groups reached on average only 21 points per graded module. Between those groups the so called „PT“ and „TT“ (see Figure 7.10) groups were created for gradually activating the access to the programming exam (PT) or the TUWEL-quiz (TT). Due to the initially higher maximum values of these groups, one can observe higher achieved points averaging between 22 and 24 points. However, by considering only the exercise groups such as „Di12b+“, „Di14a“, „Di16a“, „Di12a“, „Di18a“, and „Di14b“ no significant relations can be noted assuming that groups working in the evening perform worse.

7.3 Evaluation of the Model

The following sections intend to complete the view and assessment of the developed model and its features. This evaluation includes also comments and feedback about advantages and disadvantages of certain use cases. An additional component of this evaluation will be an expert interview, which will be used to amplify the findings or give new insights in that regard. As described in Section 7.2 the TUWEL API enables to retrieve a variety of information, which can be utilized by other visualization or BI tools.



What influences adjustedGrades to Increase ?

TimeNeeded goes up 14.34

On average when TimeNeeded increases, adjustedGrades also increases.

Figure 7.10: Average of Grades divided by TUWEL Group Names

7.3.1 Quantitative Evaluation - Statistical Analysis

To measure student performance in courses, this section represents the results of the statistical analysis. The objective is to create a statistical model, which summarizes quantitative data insights and indicates which factors influence student performance positively or negatively. According to C. Xiao et al. [41] and [97] the Spearman's correlation (see Section 2.3.2) is commonly used in the EDA domain, which suits case study of the IP1 course. Similarly, previous research of indication models by S. Liao et al. [5] and M. M. Jamjoom et al. [7] build linear regression models to determine or predict student performance in programming courses.

First of all, to keep the evaluation manageable and well-structured not all results will be explained and assessed elaborately. The evaluation will be split in two sections — TUWEL modules which are graded (described in Section 2.2.3) and contribute to the final score directly and TUWEL modules which are not graded and thus only have an indirect influence on the final score. Most of the following plots have a similar design to facilitate a convenient comparison of different attributes.

Graded Modules

The first Linear Regression Model can be observed in Figure 7.11 by comparing the programming exam grade with the final student grade. As described in the chart the student's final grade is plotted on the x axis and the programming exam result can be observed on the y axis. Furthermore, the red line indicates an estimate linear regression model, which describes the relationship between the mentioned two attributes. The green lines estimate the confidence intervals, which comprise the majority of observations or data points to get a rough visual assessment of the underlying data patterns.

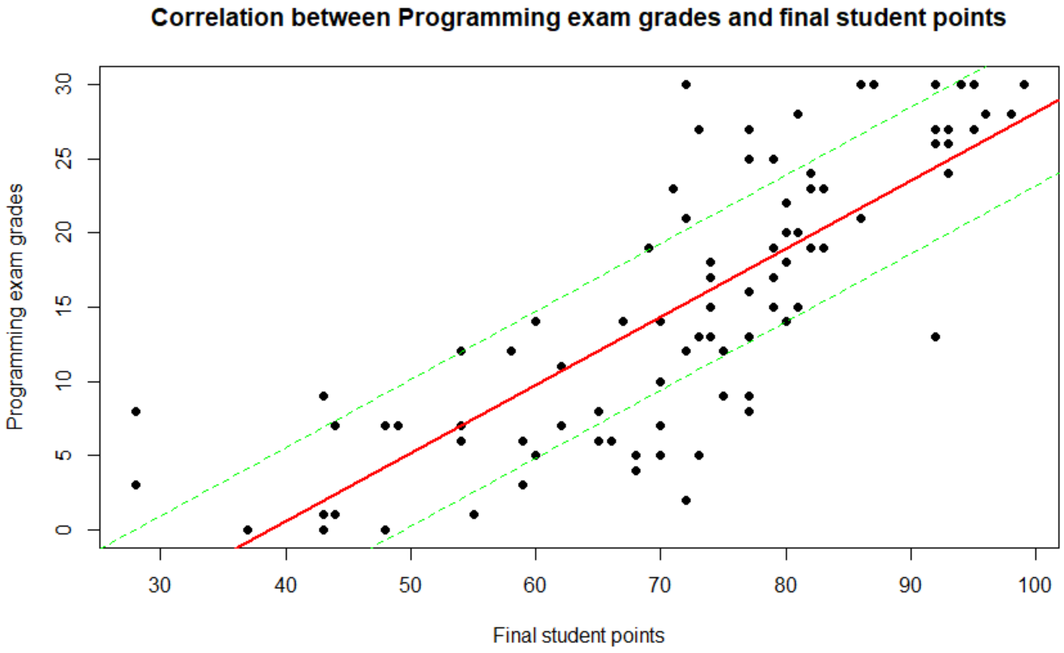


Figure 7.11: Linear Regression Model between Programming exam grades and final student grades

However, the representation in Figure 7.11 shows a full overview of the data, which does not exclude any outliers. In contrast to this chart, the Figure 7.12 displays Linear Regression using Spearman's correlation, which excludes these outliers. This chart was created using the online correlation calculator Statistics Kingdom[55].

As mentioned before, the focus of this section will be to get a basic understanding of the quantitative analysis by providing an unified visual representation. This representation will be supplemented by the evaluation results of the robust Spearman's correlation accordingly.

[55] https://www.statskingdom.com/correlation-calculator.html [last access: 12.03.2024]

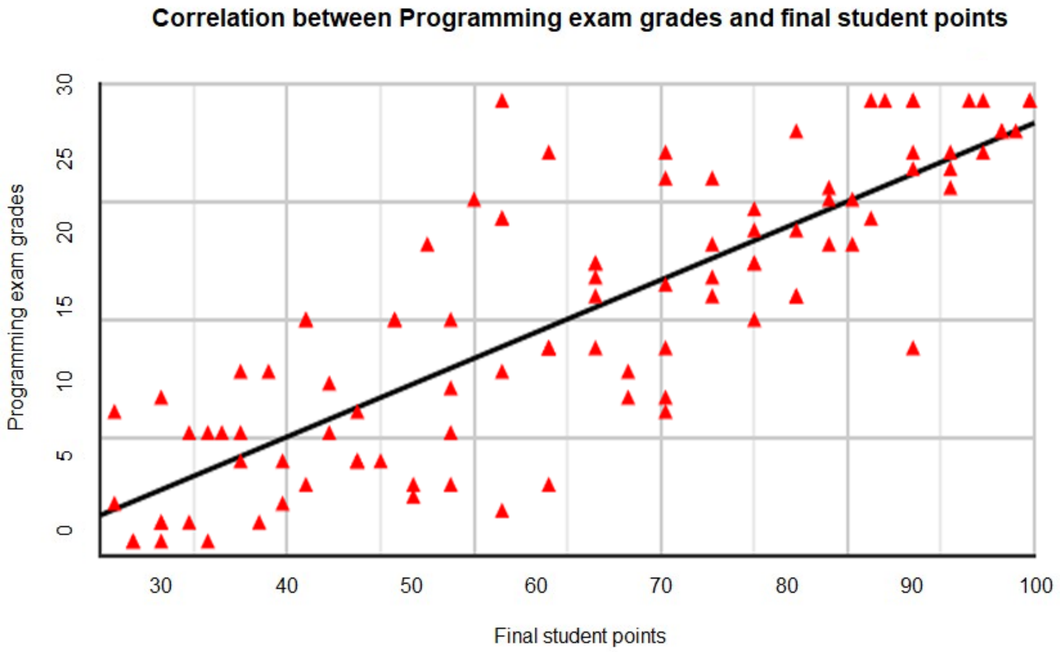


Figure 7.12: Linear Regression Model following Spearman's correlation between Programming exam grades and final student grades (zoomed)

For the comparison of the programming exam grade and the student's final grade the following report in Table 7.3 was created. The main focus of this master thesis is to determine the importance of this module, which limits the information of the mentioned table to the „Spearman's rank correlation coefficient (r_s)" and „P-value". In this case, the Spearman's correlation is positive and close to 1, which indicates a strong positive correlation (see Section 2.3.2). Furthermore, the correlation test using the p-value (described in Section 2.3.4) is equal to 0, which means that the statistical evidence of possible correlation is very strong. From these two values it can be assumed a strong positive correlation between the programming exam grades and the final student grade.

Similarly, the Linear Regression model when comparing the TUWEL exam grade and the final student grade in Figure 7.13 exhibits a strong positive correlation with a low p-value for the statistical significance. The exact values for all graded modules can be observed in Table 7.4. In general, the Table 7.4 contains the columns „Module Name", „Spearman's correlation coefficient", and „P-value". By further inspecting the table, it is clearly noticeable that almost all modules exhibit a positive correlation in respect to the final student grade attribute, which aligns with the grading scheme of the IP1 course.

Parameter Value	Value
Spearman's correlation coefficient (r)	0.8326
r^2	0.6933
P-value	0
Covariance	699.5556
Sample size (n)	100
Statistic	14.8839

Table 7.3: Output Table of the Spearman's Correlation considering programming exams and final student grade

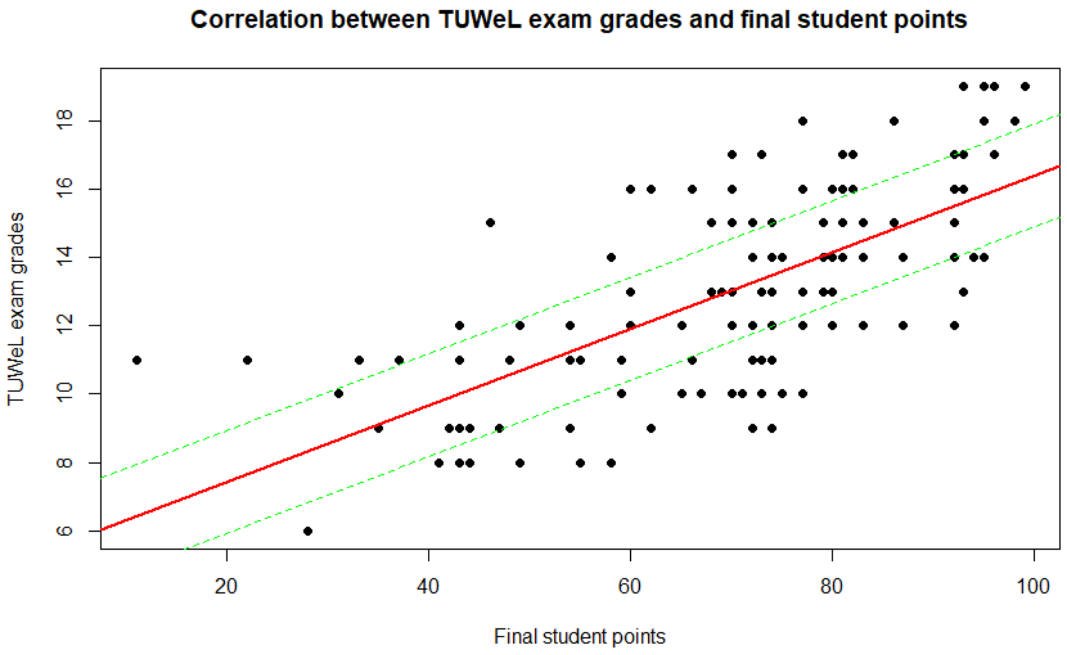


Figure 7.13: Linear Regression Model between TUWEL exam grades and final student grades

Nevertheless, the value of the Spearman's correlation coefficient is significantly lower on „Additional Collaboration Points“, „Mini-TUWEL Exam Grades“, and „Assignment Presentation Points“. This insight indicates a weaker correlation between those mentioned attributes and the final student grade. Additionally, the „P-value“ of „Additional Collaboration Points“ equals to 0.2238, which signifies a very weak or no statistical evidence of possible correlation according to the definition in Table 2.2. In other words, the correlation between Collaboration points and the final student grade is not statistically significant. This can also be seen in Figure 7.14, which displays very few and wide-apart

Module Name	Spearman's correlation coefficient	P-value
Programming Exam Grades	0.8326	0
TUWEL Exam Grades	0.6999	0
Assignment Crossed Tasks	0.6727	0
Assignment Attendance	0.6597	0
Additional Collaboration Points	0.3473	0.2238
Mini-TUWEL Exam Grades	0.2302	0.0034
Assignment Presentation Points	0.201	∼0
Exercise Points	NaN	NaN

Table 7.4: Spearman's Correlation Analysis for each graded TUWEL Module

data points or observations — thus, in this case no correlation can be ascertained by definition. The Table 7.4 also contains the special case „Exercise Points", which has Not a Number (NaN) entries. Figure 7.15 shows that all students in this course have achieved 2 points for all available exercises, which does not provide any further information and thus no correlation can be specified.

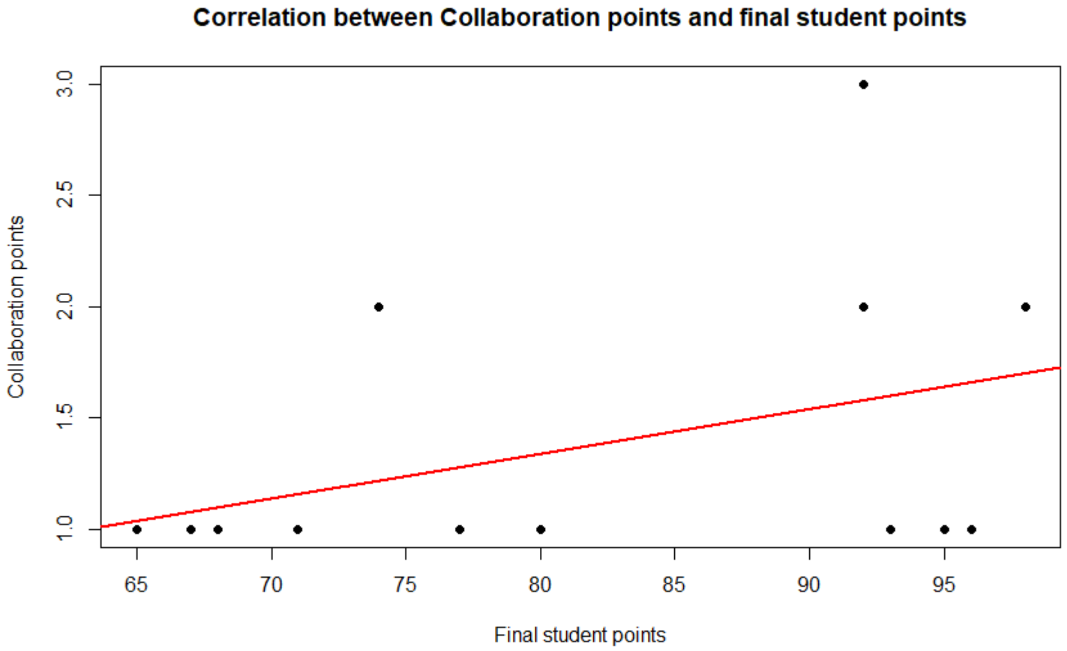


Figure 7.14: Linear Regression Model between Collaboration points and final student grades

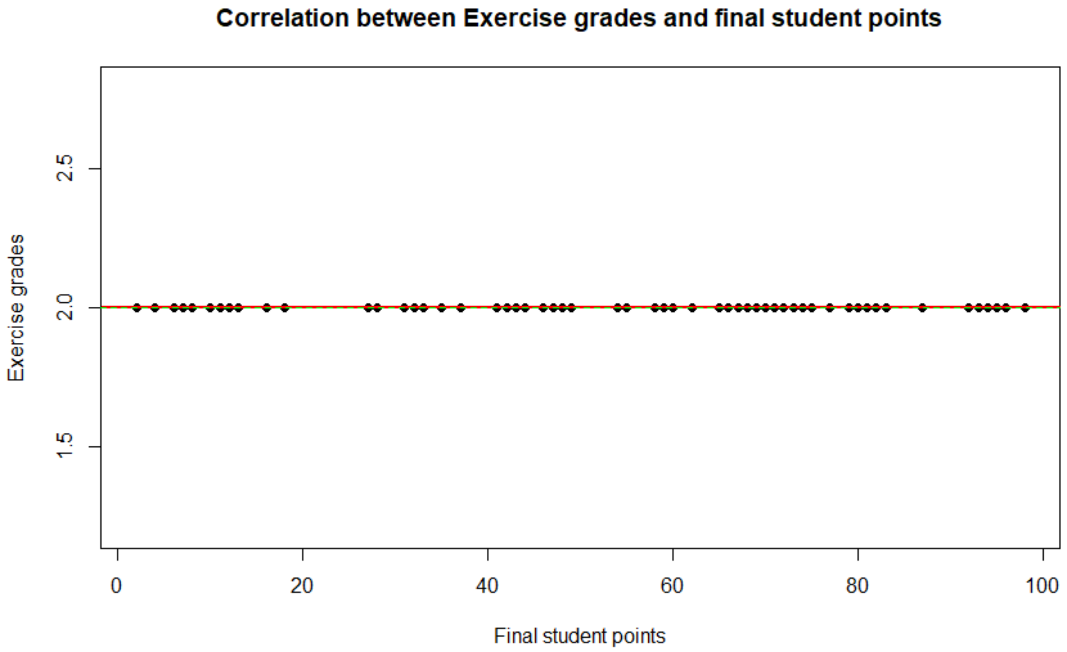


Figure 7.15: Linear Regression Model between Exercise grades and final student grades

Not Graded Modules

The second part of the quantitative analysis considers also student activities, which are not graded, but still may have an influence on the final student grade — examples would be the overall effort in example TUWEL-quizzes or „Number of Discussion Posts“ in the TUWEL forum. Similarly to graded modules, the Table 7.5 displays a summary of statistical models by using Spearman's correlation.

The first module according to the Table 7.5 is „Number of Discussion Posts“, which exhibits a fairly high Spearman's correlation coefficient. However, due to the low number of similar or structured observations as shown in Figure 7.16 the p-value is equals to 0.5594, which provides almost no statistical evidence for correlation between the number of discussion posts and final student grade.

Similar to the first module, that provides almost no statistical significance, are the following modules:

- **Mini-TUWEL Exam Time:** This module displays the correlation between the time required to finish Mini-TUWEL exams and the final student grade. Due to

Module Name	**Spearman's correlation coefficient**	**P-value**
Number of Discussion Posts	0.3536	0.5594
Matriculation Number	0.2307	0.0173
Exercise Quiz Time	0.1783	∼0
Number of Enrolled Courses 2023S	0.1758	0.0714
TUWEL Exam Time	0.1474	0.0868
Exercise Quiz Grade	0.1369	0.0018
Mini-TUWEL Exam Time	0.1008	0.1991
Submission Time	-0.0123	0.676
Number of Enrolled Courses	-0.0587	0.5502
Exercise Quiz Number of Attempts	-0.0604	0.17
Group Time Slot	-0.0837	0.3986

Table 7.5: Spearman's Correlation Analysis for each ungraded TUWEL Module

the fairly high p-value no statistical significance can be justified in this case even though the Spearman's correlation coefficient shows only a weak positive correlation between the mentioned two variables.

- **Submission Time:** This module represents the submission time of assignments in comparison to the final student grade. Interestingly, a negative Spearman's correlation coefficient portends a negative correlation — thus, the longer students require to submit an assignment the worse their overall course performance gets.

- **Number of Enrolled Courses:** This module displays the overall number of enrolled courses including the enrolled courses in the 2023S compared to the final student grade. A small negative Spearman's correlation coefficient without statistical significance can be observed in this case.

- **Exercise Quiz Number of Attempts:** This module shows the correlation between the number of attempts when submitting exercise-quizzes and the final student grades — nevertheless, very little statistical evidence of possible correlation can be noted.

- **Group Time Slot:** At last, this module describes the relationship between the time for group exercises and the final student grade — a high p-value can be noted in this instance, which precludes a statistical significance.

Moreover, the module „Matriculation number“ can be interpreted as an exception, because it represents the matriculation number of a student. The matriculation number follows a certain pattern and is automatically assigned to each and every student in the TU Wien — in most cases, student's often do not have any or only limited influence on this attribute. Noteworthily, a positive correlation using Spearman's correlation coefficient and a strong evidence for statistical significance can be ascertained, which

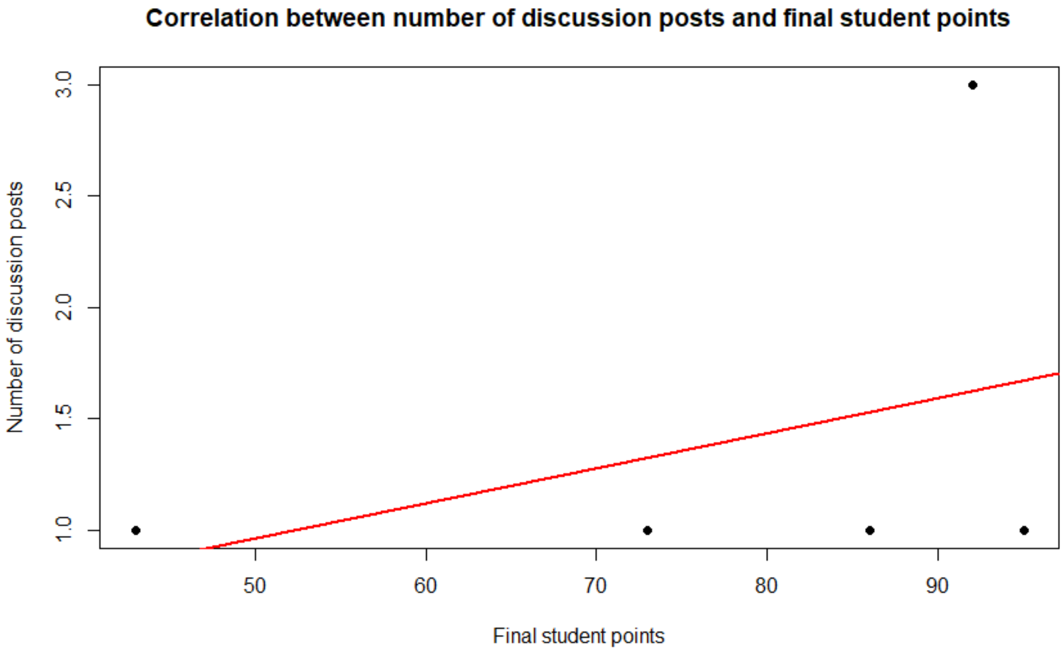


Figure 7.16: Linear Regression Model between Number of Discussion Posts and final student grades

might be an interesting finding for future work.

Another surprising insight can be observed in the modules „Exercise Quiz Time“ and „Exercise Quiz Grade“ which both feature a positive Spearman’s correlation coefficient and a strong evidence for possible statistical correlation. As seen in Figures 7.17 and 7.18 a slightly positive correlation between exercise quizzes and final student grade can be observed. The p-value of both modules indicate a very strong evidence for statistical significance and thus correlation.

The final two modules „Number of Enrolled Courses 2023S“ and „TUWEL Exam Time“ exhibit a slightly positive Spearman’s correlation coefficient, which represents a positive correlation between those mentioned modules and the final student grade. Nevertheless, for both modules the p-value is between 0.05 and 0.1, which indicates only a weak statistical evidence of possible correlation according to the Table 2.2 in Section 2.3.4. These statistical coefficients could also be mapped to the visualizations of Figures 7.19 and 7.20, which similarly only display partial correlation patterns, including a low number of observations.

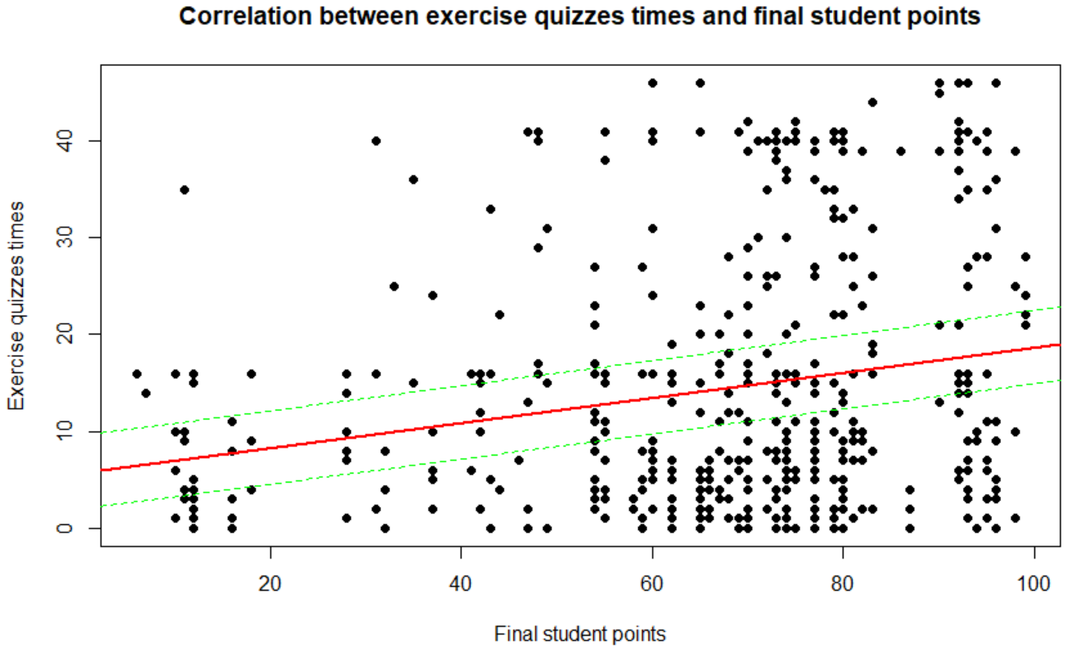


Figure 7.17: Linear Regression Model between exercise quiz times and final student grades

7.3.2 Expert Evaluation

In addition to the statistical analysis, an expert evaluation was conducted to get an overview of the current assessment of domain experts — in the case of the IP1 course these would be the course-leaders. So, the evaluation was conducted with five of the seven course-leaders and all participants were shortly introduced to this topic by email. The purpose of this inquiry is to get an approximate idea of the assessment by domain experts — in other words, due to the small number of participants no generalizable or absolute conclusion should be done without considering the quantitative analysis.

The objective of this section in this master thesis is compare and even complement the quantitative findings, which were often limited to the capabilities using the TUWEL API. For this purpose, the following list containing questions about all modules, quizzes, resources, and additional factors, which could have an influence on the student's performance, was created:

I How influential is the **forum-activity** of students on the course performance?

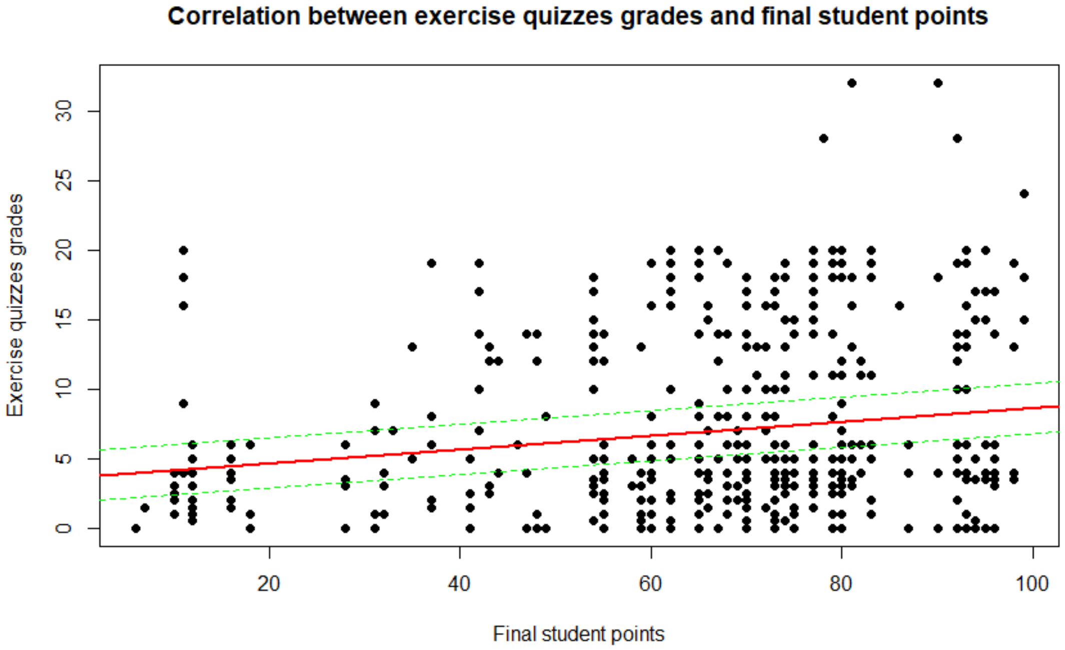


Figure 7.18: Linear Regression Model between exercise quiz grades and final student grades

II How influential is the **number of enrolled courses** of students on the course performance?

III How influential is the **enrolled group (EP1 exercise-groups)** of students on the course performance?

IV How influential is the **effort/performance in test-quizzes** of students on the course performance?

V How influential is the **number of attempts in test-quizzes** of students on the course performance?

VI How influential is the **access or download time of resources** of students on the course performance?

VII How influential is the **achieved points in assignments** of students on the course performance?

VIII How influential is the **presenting skills/performance of assignments** of students on the course performance?

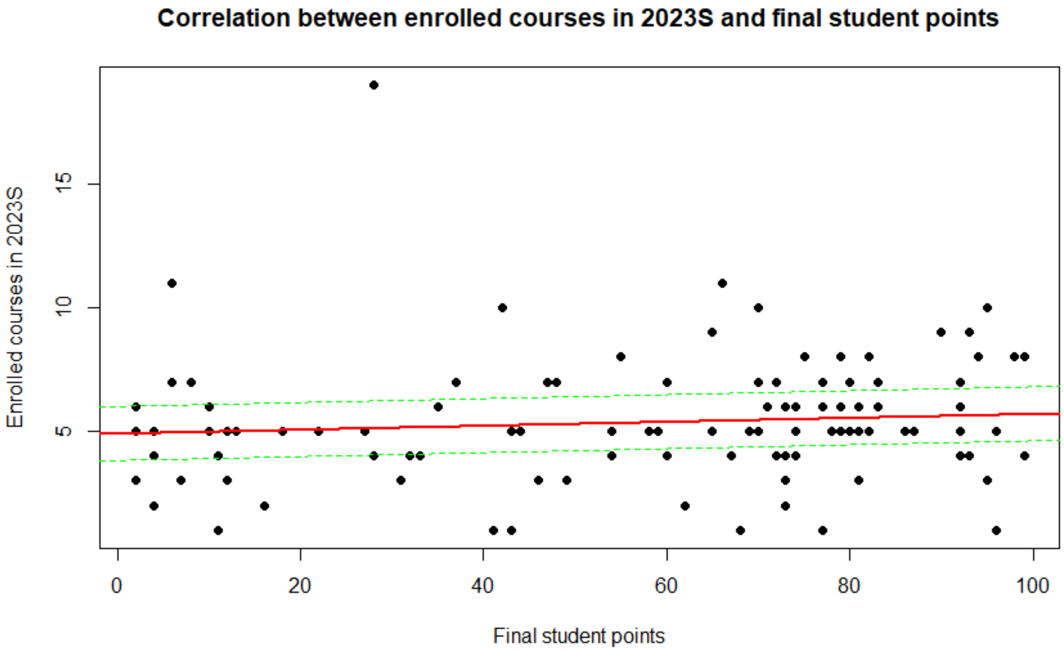


Figure 7.19: Linear Regression Model between enrolled courses in 2023S and final student grades

IX How influential is the **effort/performance on non-graded assignments** of students on the course performance?

X How influential is the **previous knowledge (considering K2- and K3-groups)** of students on the course performance?

XI How influential is the **test-exam attempts** of students on the course performance?

Each question could be rated by five domain experts from one (Very Influential) to ten (Not Influential) based on their personal experience in the IP1 course. An important pre-requisite was that no participants in this questionnaire conducted any prior analysis of the IP1 course, which might affect their personal assessment. The results of the expert evaluation using box plots can be seen in Figure 7.21, which show that most of the specified items display a score of six or lower. Thus, the majority of participating experts think, that these mentioned items could also be influential when considering the student performance and latter final student grade.

Lastly, at the end of the questionnaire an open question was implemented to ask for additional factors influencing the student performance. The following answers were

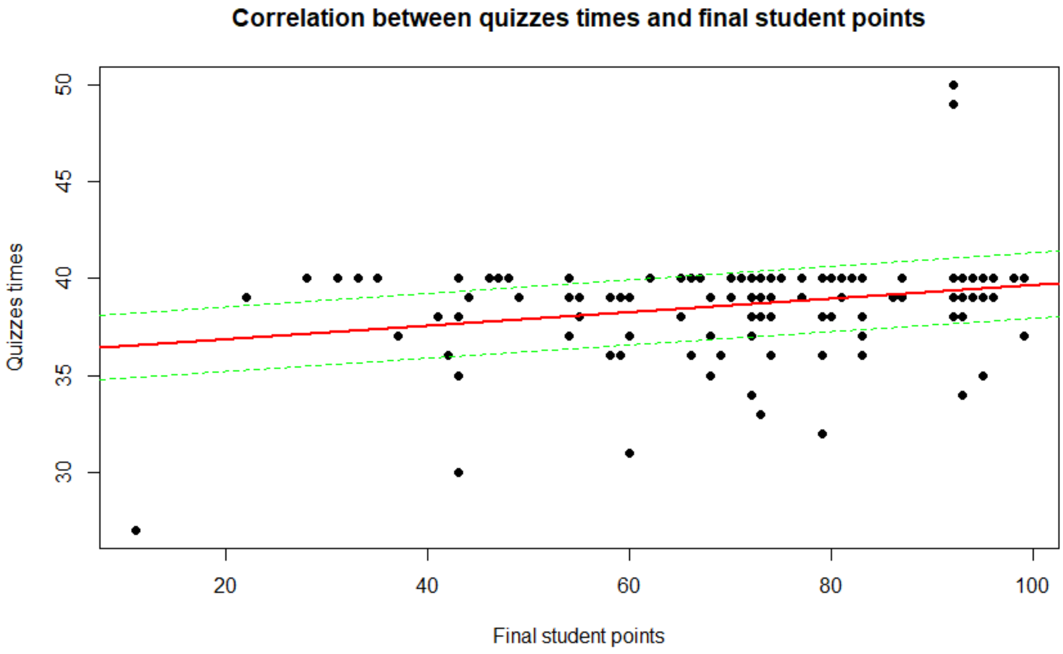


Figure 7.20: Linear Regression Model between TUWEL exam times and final student grades

provided by the domain experts, which extends the prior list to be used in the final Section 8 of this master thesis:

XII Student's semester design — for instance is this semester very stressful or relaxing for the student to be able to concentrate on the IP1 course.

XIII Availability of third-party sources to copy solutions from peers, which could have a negative impact on the student's performance (even though the assignments are submitted without flaws).

XIV The intrinsic motivation and ambition of the student.

XV Social handling when submitting assignments or requiring help from peers or university staff.

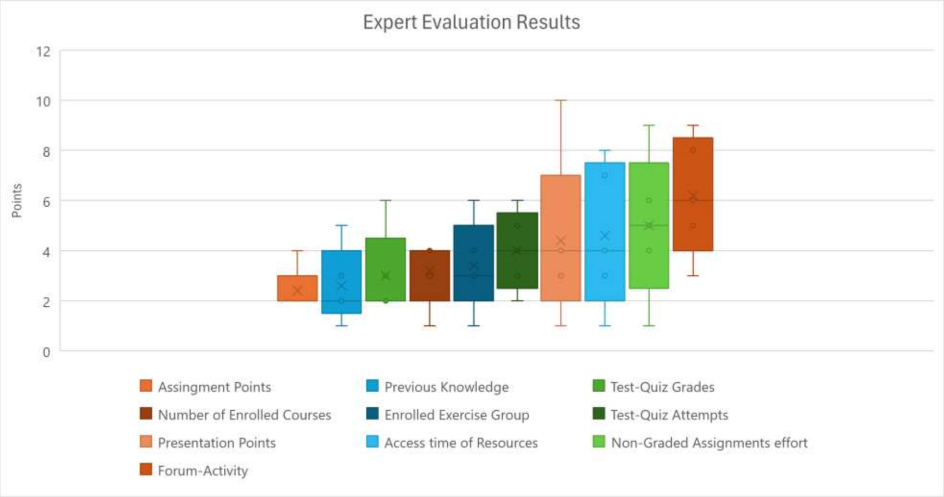


Figure 7.21: Results of Expert Evaluation from Very Influential (1) to Not Influential (10)

CHAPTER 8

Discussion and Summary

This final chapter discusses the development process of the introduced model and its features, insights, and further possible advancements in Section 8.1. This section sets the foundation to answer the research questions (see Section 8.2) previously defined in Section 1.1. Lastly, a final closing Statement, which includes a summary and future outlook will be provided in Section 8.3.

8.1 Interpretation and Comparison of the Evaluation Results

The advancements of using the visualizations and further statistical model might be quite beneficial for university staff to allocate resources more efficiently. In this section of this master thesis possible further adjustments and capabilities will be discussed in more detail. Secondly, a comparison between the quantitative analysis and expert evaluation could bring additional insights or set a base for future work.

After consultation with Dr. S. Podlipnig a number of improvements were discovered, which might improve the results of the conducted quantitative analysis. First of which considers test-quizzes, which could contain practice attempts as students finish the exam within a few minutes. The solutions are often used to improve their test-quiz performance on further tries and falsely indicate their improvement over time. However, the developed model was designed to be used on other courses, which makes it quite hard to include specific rules or thresholds, when a test-quiz was submitted truthfully. In addition to that, it is possible that students might collaborate among themselves which could also shift the actual result of their performance.

Module Name	Spearman's correlation coefficient	P-value
Programming Exam Grades	0.8326	0
TUWEL Exam Grades	0.6999	0
Assignment Crossed Tasks	0.6727	0
Assignment Attendance	0.6597	0
Matriculation Number	0.2307	0.0173
Mini-TUWEL Exam Grades	0.2302	0.0034
Assginment Presentation Points	0.201	∼0
Exercise Quiz Time	0.1783	∼0
Exercise Quiz Grade	0.1369	0.0018

Table 8.1: Spearman's Correlation Analysis for each TUWEL Module

Another flaw, which is quite hard to prevent are diverse Moodle errors or faults, which affect the submission time of modules, quizzes, and other resources. That could also be the reason, why some charts such as Figure 7.20 contain some extreme outliers — by using the Spearman's correlation this effect can be reduced or even avoided completely.

Furthermore, when comparing courses over several semesters possible deviations could occur due to slight changes in the grading scheme, learning focus, and difficulty of assignments, quizzes, and exams. Due to the lack of recorded data or information about these deviations and external factors, the model must be adapted to avoid shifts in the observations, which could lead to inaccurate results. Also, depending on the number of students and the impact of the adjusted grading item, some gaps and thus shifts in statistical data could be noticed. These events and changes in the curriculum should be questioned in consultation with domain experts such as the course-leaders — to run ahead, this model is no standalone solution, due to these mentioned flaws and external factors.

8.1.1 Comparison of the Evaluations

To ensure an easier comparison between the quantitative analysis and the expert evaluation one table was created for each evaluation method. The Table 8.1 was sorted according to Spearman's correlation coefficient and p-values higher than 0.05 were excluded in this view, due to weak or no statistical evidence for possible correlation between various modules and the final student grades. The Table 8.2 was sorted according to the average points by starting with low values, which represent „Very Influential“ factors.

As shown in Table 8.1 most of the modules, which strongly influence the student's final grade are graded modules (see Section 7.3.1) — the ungraded modules „Matriculation Number“, „Exercise Quiz Time“, and „Exercise Quiz Grade“ do also show strong evidence

Factor Name	Average Points
Assignment Points	2.4
Previous Knowledge	2.6
Exam Quiz Grades	3
Number of Enrolled Courses	3.2
Enrolled Exercise Group	3.4
Test-Quiz Attempts	4
Assignment Presenting Points	4.4
Access time of Resources	4.6
Non-graded Assignments effort	5
Forum-Activity	6.2

Table 8.2: Expert evaluation average points by course-Leaders

to influence the student grade positively. By comparing the factors with the expert evaluation a certain pattern can be observed — „Assignment Points“, „Exam Quiz Grades“, and „Assignment Presenting Points“ are also quoted as „Influential“. Some factors such as the „previous knowledge“ could not be generally tested due to missing professional student information before participating in the IP1 course. Unfortunately, other factors such as „Access time of Resources“ could not be measured using the statistical model — nevertheless, the expert evaluation gives rough estimations how beneficial an early start would be to achieve better grades in assignments, exams, and exercises.

Furthermore, the factors „Number of Enrolled Courses“, „Enrolled Exercise Group“, and „Test-Quiz Attempts“ could be measured using the statistical model. Even though, these mentioned factors indicated a slight correlation using Spearman’s correlation no statistical significance could be ascertained in the summer term of 2023 of the IP1 course. Thus, following the rule by the Barcelona Field Studies Centre S.L. [47] no evidence for a possible correlation of the student data will be accepted for those course-activities.

To summarize these findings and the comparison of the evaluation methods, the course-leaders already had a good understanding and assessment of essential factors influencing the final student grade. Still, the developed statistical model provides a data-based ranking how important each module or factor is. A combination of the developed model and visualization and additional expertise by domain experts would be a suitable solution to indicate student performance and thus effectively align resources according to these insights.

8.2 Answering the Research Questions

As mentioned above, this section summarizes the gathered information from the initial semi-systematic literature review combined with the evaluation results to answer the research questions (see Section 1.1) of this master thesis — each question will be addressed using „re“.

re RQ1: *To what extend can students at risk of failing a course be determined by previous activities?*
For reference, this question was developed to examine the capabilities and also limits of a quantitative analysis using the LMS TUWEL. In fact, there is a number of ways to track student performance using different measures such as quoted in previous research by C. Gordon et al. [4], who talked about various data collections methods. These methods reach from giving surveys, gathering demographic data, and conducting questionnaires in class to more advanced model or tools to predict student performance using statistical models. Research by A. K. Veerasamy et al. [61], E. M. Queiroga et al. [13], M. M. Jamjoom et al. [7], and S. N. Liao et al. [5] focus on various methods and early assessments, which are described in more detail in Sections 3.1.1 and 3.3.

Additional results can be observed by considering the Design Science approach building a statistical representation to model a course's data using the management tool TUWEL. Especially by contemplating the results in Section 7.3, clear correlations between in-course activities and the final student performance can be noticed. The exact effects of certain activities can be viewed in Section 8.1. Nevertheless, previous activities, particularly graded items, directly determine the student's final grade — according to domain experts also intrinsic values and the student's ambition might have an impact on the overall performance. Unfortunately, it is hardly possible to provide an estimation or exact number due to many different internal and external factors, which includes the lecture content itself as well. The developed visualization and statistical analysis combined with expert knowledge can reduce the uncertainty.

re RQ2: *Which combination of attributes is most relevant to approximate a student's final score?*
This research question is closely related to RQ1, because it examines essential activities which are required to obtain a positive course grade. Thus, the final combination of attributes or activities depends on many different factors and additional influences, which are sometimes not even alterable by students such as the difficulty of certain exam-slots or group performances. However, the developed statistical analysis of TUWEL-courses can provide further insights into essential activities, which directly influence the final student grade. As within the case study of the IP1 course also non-graded quizzes had a positive impact on the student's performance, which can be observed in Section 7.3.1. Ultimately, by combing the expert evaluation, insights of scientific literature, and quantitative results the following combination of attributes are relevant to approximate the student's grade in the IP1 course:

- Programming Exam Grade

- TUWEL Exam Grade
- Assignment Grade, which includes the attendance during exercise-sessions and the student's presentation performance
- Matriculation number (this attribute is not alterable, but statistically gives insights about the student's final grade)
- Mini-TUWEL Exam Grade
- Exercise Quiz performance or effort
- Previous (programming) knowledge
- External and internal factors such as intrinsic motivation

Generally speaking, the application and implementation of the developed visualizations and statistical models can be used on various courses by following the detailed documentation provided in this master thesis. As mentioned before, it is highly recommended to combine the quantitative results with expert knowledge to avoid false conclusions due to unknown internal and external effects.

re RQ3: *Which attributes or properties of course-contents or taught programming skills are hard to comprehend for computer science students?*
Lastly, the purpose of this research question is to find or determine sections of a programming course, in which student require extensive support by tutors, lecturers, or trainers. According to previous research by B. Özmen and A. Altun [98] the biggest cause for failure in programming languages is due to lack of knowledge and practice. S. Derus and A. Z. Ali [99] share these findings, while programming students often have issues transforming abstract problems into code — the lack of visual feedback and lack of active involvement could be reasons for this behavior. The researchers suggest more active discussion about code solutions, which is already given in the case study of the IP1 course.

A very promising qualitative analysis was conducted by E. Lahtinen et al. [100] in 2005, which split programming concepts to determine „difficult to learn“ course contents for novice programmers. Unfortunately, it is quite hard to determine difficult course contents using the case study of the IP1 course. The reason for that is described in Section 6.1, which briefly states that it is not possible to split programming concepts of each exercise or assignment effectively due to the structure of the IP1 course. This aspect might also be considered and examined in future work by testing the developed visualizations and statistical model on more suitable programming courses — the IP1 course constitutes a good example, that the model requires detailed grading information regarding programming concepts to precisely determine difficult programming topics. Nevertheless, by comparing the previous research with the results of the quantitative analysis considering solely descriptive statistical values, the following course contents might cause the most issues among programming beginners:

- Recursion
- Structured data types
- Abstract data types
- Input and output handling
- Usage of external libraries or code

Analogously to the previous research questions, external influences or changes in course contents could cause deviations in the quantitative analysis. To enhance the precision of this analysis within the developed statistical model in programming courses, supplementary mapping information is required, which may be the focal point of future work.

8.3 Closing Statement

In conclusion, this master thesis investigated two scientific fields: EDM and Data Visualization with the ultimate goal to determine student performance in programming courses using LMSs. Utilizing this information would not only benefit students, which might receive extensive support, but also improve the university's resource allocation including student staff. Even though, the management platform Moodle (see Section 3.2.1) is one of the most widely used LMSs in Europe only little further research was conducting in this domain. A reason for that could be, that Moodle already provided a number of analysis and visualization methods, which were mostly interesting for very large courses.

Many courses at the TU Wien are managed using the Moodle instance TUWEL, which was adapted and maintained to the needs of the university. In other words, close to no scientific information was conducted using this management platform. Similarly, TUWEL also provided some additional analysis methods for its hosted courses, which were in some cases limited to Moodle's generic design.

In close consultation with the TUWEL technical team and Dr. S. Podlipnig a generic model was developed to fill in these gaps regarding analysis and evaluation without requiring additional tracking or monitoring of student behavior. The only information source, which is required by the tool is the mentioned management platform itself. This characteristic of the model limited its effectiveness on the data saved in TUWEL, which fortunately was not an issue when inspecting the IP1 course.

So, the first step of this master thesis was a semi-systematic literature review to get an understanding, how to develop such a tool and compare the solution with existent experiments or literature. This part of this master thesis was also essential to answer the specified research question in Section 8.2. Based on this research and prior experience, the course information was fetched using TUWEL's API by following the ETL approach described in Section 5. The interactive visual representations were created using Power BI and its features to model the data and gain first insights into the course or student data.

The feedback and expertise provided by Dr. S. Podlipnig helped improving the results of the visualizations and statistical model by considering the peculiarities of the IP1 course. At the end of the development process, two analysis methods were established to combine the statistical results with an expert evaluation, which increases the validity of the case study and further the developed model.

8.3.1 Future Work

Nevertheless, due to the study's time constraints, not all extensions and possible analysis or evaluation ideas by domain experts could be realized. Therefore, this master thesis provides a solid foundation for future work in the following domains.

One of the approaches for future work might be to consider utilizing this model and its features also for non-programming courses. This proposition includes adapting the visualization tool and the developed statistical model to fit more extensive use cases, thus courses.

As this model's processes are not fully automated, there is currently no real-time analysis of courses possible. By improving the used algorithms or including parallelization the execution time (see Section 7.1) can be significantly shortened. Depending on the use case, this addition might be very useful for live-tracking students in quizzes or university staff receiving direct feedback after examinations.

Furthermore, some aspects, such as previous programming knowledge or intrinsic motivation cannot be examined using the developed model. However, combining its quantitative findings with these theoretical insights might further increase the effectiveness and accuracy of the model. In other words, this master thesis could represent a solid foundation for further research in this domain. In addition to that, examining the psychological effects of assessing students during an ongoing semester might bring new valuable insights as well.

Closely related to the previous idea is the implementation of a classification or prediction model to predict student information during the ongoing course. This was also the initial idea of this master thesis, which was put away due to the high complexity of creating a generic prediction tool while considering also external influences. A conceivable attempt would be to record student information in greater detail, which might result into more exact conclusions. This idea would have exceeded the scope of this work — still, this master thesis could set the foundation for such an extension.

Lastly, as mentioned in Section 3.3, the management platform TUWEL uses specific extensions such as an anonymous forum, which currently contains no API-support. By integrating this feature, additional insights regarding forum activities can be conducted using the developed model of this master thesis. Supplementary, further research regarding course submissions and submission time of students and its effects on the final student grade could additionally convey valuable outcomes.

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 32

API Application Programming Interface. 12, 13, 27, 28, 41, 43, 44, 46, 62, 66, 69, 78, 89, 90

BI Business Intelligence. 28, 29, 31, 32, 38, 41, 55, 69

Blackboard LMS Blackboard Learning Management System. 26, 27

CEEB College Entrance Examination Board. 16

CPU Central Processing Unit. 61

CSE Computer and Information Science Education. 23

CSV Comma Separated Values. 13, 39, 46, 51

DAX Data Analysis Expression. 47

DVCS Distributed Version Control System. 10, 12

ECTS European Credit Transfer System. 15

EDA Exploratory Data Analysis. 19, 70

EDM Educational Data Mining. 1–3, 5–7, 23–26, 32, 34, 62, 88

ETL Extraction-Transformation-Loading. 41, 51, 55, 89

GUI Graphic User Interface. 13, 27, 51

HEO Higher Education Organization. 26

IP1 Introduction to Programming 1. 5, 7, 9, 14, 15, 17, 33, 35, 36, 39, 46, 51, 55, 57–66, 70, 72, 78, 80, 81, 85–89, 93

LMS Learning Management System. 6, 7, 26–28, 32, 35, 62, 86, 88

NaN Not a Number. 74

OLS Ordinary Least Squares. 20–22

OOP Object-oriented Programming. 10

OS Operating System. 61

OSI Open Source Initiative. 27

Power BI Power Business Intelligence. 13, 14, 32, 38, 39, 41, 46, 47, 49–51, 57, 62, 89

RAM Random-Access Memory. 61

RCS Revision Control System. 10

SQL Structured Query Language. 31, 46

TU Wien Technische Universität Wien. xi, xv, 9, 12, 14, 15, 28, 33, 39, 46, 57, 67, 76, 88

TUWEL Technische Universität Wien E-Learning Platform. 4, 12, 13, 26, 28, 33, 34, 39, 41–44, 46, 50, 59, 60, 62, 64–70, 72, 75, 78, 86–90

VCS Version Control System. 10

Bibliography

Print Resources

[1] Simon, A. Luxton-Reilly, V. V. Ajanovski, *et al.*, „Pass rates in introductory programming and in other STEM disciplines“, *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, pp. 53–71, 2019, ISSN: 1942647X. DOI: 10.1145/3344429.3372502.

[3] J. Wolff, „How Is Technology Changing the World, and How Should the World Change Technology?“, *Global Perspectives*, vol. 2, no. 1, p. 27 353, Aug. 2021, ISSN: 2575-7350. DOI: 10.1525/gp.2021.27353.

[4] C. Gordon, S. Zhao, and F. Vahid, „Ultra-Lightweight Early Prediction of At-Risk Students in CS1“, *SIGCSE 2023 - Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, vol. 1, pp. 764–770, 2023. DOI: 10.1145/3545945.3569764.

[5] S. N. Liao, D. Zingaro, M. A. Laurenzano, W. G. Griswold, and L. Porter, „Lightweight, early identification of at-risk CS1 students“, in *ICER 2016 - Proceedings of the 2016 ACM Conference on International Computing Education Research*, Association for Computing Machinery, Inc, 2016, pp. 123–131, ISBN: 9781450344494. DOI: 10.1145/2960310.2960315.

[6] S. Bergin, R. Reilly, and D. Traynor, „Examining the role of self-regulated learning on introductory programming performance“, *Proceedings of the 1st International Computing Education Research Workshop, ICER 2005*, pp. 81–86, 2005. DOI: 10.1145/1089786.1089794.

[7] M. M. Jamjoom, E. A. Alabdulkareem, M. Hadjouni, F. K. Karim, and M. A. Qarh, „Early prediction for at-risk students in an introductory programming course based on student self-efficacy“, *Informatica (Slovenia)*, vol. 45, no. 6, 2021, ISSN: 18543871. DOI: 10.31449/INF.V45I6.3528.

[9] S. B. Kotsiantis, C. J. Pierrakeas, and P. E. Pintelas, „Preventing student dropout in distance learning using machine learning techniques“, *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 2774 PART 2, no. September, pp. 267–274, 2003, ISSN: 03029743. DOI: 10.1007/978-3-540-45226-3_37.

[10] K. Bunkar, U. K. Singh, B. Pandya, and R. Bunkar, „Data mining: Prediction for performance improvement of graduate students using classification“, *IFIP International Conference on Wireless and Optical Communications Networks, WOCN*, pp. 1–5, 2012, ISSN: 21517681. DOI: `10.1109/WOCN.2012.6335530`.

[11] P. Ihantola, A. Vihavainen, A. Ahadi, *et al.*, „Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies“, in *Proceedings of the 2015 ITiCSE on Working Group Reports*, ser. ITICSE-WGR ’15, Vilnius, Lithuania: Association for Computing Machinery, 2015, 41–63, ISBN: 9781450341462. DOI: `10.1145/2858796.2858798`. [Online]. Available: `https://doi.org/10.1145/2858796.2858798`.

[12] A. K. Veerasamy, M. J. Laakso, and D. D’Souza, „Formative Assessment Tasks as Indicators of Student Engagement for Predicting At-risk Students in Programming Courses“, *Informatics in Education*, vol. 21, no. 2, pp. 375–393, 2022, ISSN: 16485831. DOI: `10.15388/infedu.2022.15`.

[13] E. M. Queiroga, M. F. Batista Machado, V. R. Paragarino, T. T. Primo, and C. Cechinel, „Early Prediction of At-Risk Students in Secondary Education: A Countrywide K-12 Learning Analytics Initiative in Uruguay“, *Information (Switzerland)*, vol. 13, no. 9, 2022, ISSN: 20782489. DOI: `10.3390/info13090401`.

[20] N. N. Zolkifli, A. Ngah, and A. Deraman, „Version Control System: A Review“, *Procedia Computer Science*, vol. 135, pp. 408–415, 2018, ISSN: 18770509. DOI: `10.1016/j.procs.2018.08.191`.

[23] O. M. Khanday and S. Dadvandipour, „Doktoranduszok Fóruma“, 2020. [Online]. Available: `https://www.researchgate.net/publication/343418314_Doktoranduszok_Foruma`.

[29] L. T. Becker and E. M. Gould, „Microsoft Power BI: Extending Excel to Manipulate, Analyze, and Visualize Diverse Data“, *Serials Review*, vol. 45, no. 3, pp. 184–188, 2019, ISSN: 00987913. DOI: `10.1080/00987913.2019.1644891`.

[31] S. Loskovska, „The Review and Introduction of ECTS System“, *2nd Tempus JEP Workshop*, 2008.

[32] J. Schneider and E. Hutt, „Making the grade: a history of the A-F marking scheme“, *Journal of Curriculum Studies*, vol. 46, no. 2, pp. 201–224, 2014, ISSN: 00220272. DOI: `10.1080/00220272.2013.790480`.

[33] H. Kirschenbaum, R. Napier, and S. B. Simon, „Wad-Ja-Get? The Grading Game in American Education.“, 2021. DOI: `https://doi.org/10.3998/mpub.11900733`.

[34] R. Boleslavsky and C. Cotton, „Grading standards and education quality“, *American Economic Journal: Microeconomics*, vol. 7, no. 2, pp. 248–279, 2015, ISSN: 19457685. DOI: `10.1257/mic.20130080`.

[37] S. Few, *Dual-scaled axes in graphs—are they ever the best solution*, 2008. [Online]. Available: https://www.perceptualedge.com/articles/visual_business_intelligence/dual-scaled_axes.pdf.

[38] P. Isenberg, A. Bezerianos, P. Dragicevic, and J.-D. Fekete, „A Study on Dual-Scale Data Charts“, *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2469–2478, 2011. DOI: 10.1109/TVCG.2011.160.

[39] N. J. Gogtay and U. M. Thatte, *Principles of Correlation Analysis*, 2017. [Online]. Available: https://www.kem.edu/wp-content/uploads/2012/06/9-Principles_of_correlation-1.pdf.

[41] C. Xiao, J. Ye, R. M. Esteves, and C. Rong, „Using Spearman's correlation coefficients for exploratory data analysis on big dataset“, 2015. DOI: https://doi.org/10.1002/cpe.3745.

[42] A. F. Zuur, E. N. Ieno, and C. S. Elphick, „A protocol for data exploration to avoid common statistical problems“, *Methods in Ecology and Evolution*, vol. 1, no. 1, pp. 3–14, 2010. DOI: 10.1111/j.2041-210x.2009.00001.x.

[44] M. L. Dion, „Teaching ordinary least squares regression“, *Political Science and Public Policy 2022*, pp. 134–142, 2022. DOI: https://doi.org/10.4337/9781800885288.00026.

[45] A. Gelman and H. Stern, „The difference between "significant" and "not significant" is not itself statistically significant“, *American Statistician*, vol. 60, no. 4, pp. 328–331, 2006, ISSN: 00031305. DOI: 10.1198/000313006X152649.

[48] M. S. Chen, J. Han, and P. S. Yu, „Data mining: An overview from a database perspective“, *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 866–883, 1996, ISSN: 10414347. DOI: 10.1109/69.553155.

[49] I. Koprinska, J. Stretton, and K. Yacef, „Students at Risk : Detection and Remediation“, *Proceeding of the 8th International Conference on Educational Data Mining, EDM15*, pp. 512–515, 2015.

[50] S. K. Mohamad and Z. Tasir, „Educational Data Mining: A Review“, *Procedia - Social and Behavioral Sciences*, vol. 97, pp. 320–324, 2013, ISSN: 18770428. DOI: 10.1016/j.sbspro.2013.10.240.

[51] E. B. Costa, B. Fonseca, M. A. Santana, F. F. de Araújo, and J. Rego, „Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses“, *Computers in Human Behavior*, vol. 73, pp. 247–256, 2017, ISSN: 07475632. DOI: 10.1016/j.chb.2017.01.047.

[52] J. Kay, N. Maisonneuve, and K. Yacef, „Mining Patterns of Events in Students' Teamwork Data“, *In Educational Data Mining Workshop, held in conjunction with Intelligent Tutoring Systems (ITS*, pp. 45–52, 2006. [Online]. Available: https://www.educationaldatamining.org/ITS2006EDM/Kay_Yacef.pdf.

[53] L. Talavera and E. Gaudioso, „Mining student data to characterize similar behavior groups in unstructured collaboration spaces“, *Proceedings of Workshop on Artificial intelligence in CSCL*, pp. 17–23, 2004, ISSN: 1041-4347. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=12fd4b8d22052875064d43b6a7c4cfcf7f499872.

[54] D. Perera, J. Kay, I. Koprinska, K. Yacef, and O. R. Zaane, „Clustering and sequential pattern mining of online collaborative learning data“, *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 6, pp. 759–772, 2009, ISSN: 10414347. DOI: 10.1109/TKDE.2008.138.

[55] C. Carmona, G. Castillo, and E. Millán, „Discovering student preferences in e-learning“, *CEUR Workshop Proceedings*, vol. 305, pp. 33–42, 2007, ISSN: 16130073. [Online]. Available: https://www.researchgate.net/publication/228356097_Discovering_Student_Preferences_in_E-Learning.

[56] M. Pechenizkiy, T. Calders, E. Vasilyeva, and P. De Bra, „Mining the student assessment data: Lessons drawn from a small scale case study“, *Educational Data Mining 2008 - 1st International Conference on Educational Data Mining, Proceedings*, pp. 187–191, 2008. [Online]. Available: https://www.researchgate.net/publication/221570394_Mining_the_Student_Assessment_Data_Lessons_Drawn_from_a_Small_Scale_Case_Study.

[57] M. Blagojević and Ž. Micić, „A web-based intelligent report e-learning system using data mining techniques“, *Computers and Electrical Engineering*, vol. 39, no. 2, pp. 465–474, 2013, ISSN: 00457906. DOI: 10.1016/j.compeleceng.2012.09.011. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0045790612001772?via%3Dihub.

[58] W. He, „Examining students’ online interaction in a live video streaming environment using data mining and text mining“, *Computers in Human Behavior*, vol. 29, no. 1, pp. 90–102, 2013, ISSN: 07475632. DOI: 10.1016/j.chb.2012.07.020.

[59] Y.-C. Shih, P.-R. Huang, Y.-C. Hsu, and S. Y. Chen, „A COMPLETE UNDERSTANDING OF DISORIENTATION PROBLEMS IN WEB-BASED LEARNING“, vol. 11, no. 3, pp. 1–13, 2012. [Online]. Available: https://files.eric.ed.gov/fulltext/EJ989194.pdf.

[60] A. Anjewierden, B. Kolloffel, and C. Hulshof, „Towards educational data mining: Using data mining methods for automated chat analysis to understand and support inquiry learning processes“, in *International Workshop on Applying Data Mining in e-Learning (ADML 2007)*, 2007. [Online]. Available: https://telearn.hal.science/hal-00190067/document.

[61] A. Kumar Veerasamy, D. D’Souza, M. V. Apiola, M. J. Laakso, and T. Salakoski, „Using early assessment performance as early warning signs to identify at-risk students in programming courses“, *Proceedings - Frontiers in Education Conference, FIE*, vol. 2020-Octob, 2020, ISSN: 15394565. DOI: 10.1109/FIE44824.2020.9274277.

[62] J. Skalka and M. Drlik, „Automated assessment and microlearning units as predictors of at-risk students and students' outcomes in the introductory programming courses“, *Applied Sciences (Switzerland)*, vol. 10, no. 13, 2020, ISSN: 20763417. DOI: 10.3390/app10134566.

[63] A. A. Mubarak, H. Cao, and W. Zhang, „Prediction of students' early dropout based on their interaction logs in online learning environment“, *Interactive Learning Environments*, 2020, ISSN: 17445191. DOI: 10.1080/10494820.2020.1727529.

[64] S. S. Ajibade and A. Adediran, „An overview of big data visualization techniques in data mining“, *International Journal of Computer Science and Information Technology Research*, vol. 4, no. 3, pp. 105–113, 2016, ISSN: 2348-120X. [Online]. Available: https://www.researchgate.net/publication/305905594_An_Overview_of_Big_Data_Visualization_Techniques_in_Data_Mining.

[65] S. Gama and D. Goncalves, „Visualizing large quantities of educational datamining information“, *Proceedings of the International Conference on Information Visualisation*, no. July, pp. 102–107, 2014, ISSN: 10939547. DOI: 10.1109/IV.2014.65.

[67] S. Slater, S. Joksimović, V. Kovanovic, R. S. Baker, and D. Gasevic, „Tools for Educational Data Mining: A Review“, *Journal of Educational and Behavioral Statistics*, vol. 42, no. 1, pp. 85–106, 2017, ISSN: 19351054. DOI: 10.3102/1076998616666808.

[68] R. Paiva, I. I. Bittencourt, W. Lemos, A. Vinicius, and D. Dermeval, „Visualizing learning analytics and educational data mining outputs“, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10948 LNAI, no. June, pp. 251–256, 2018, ISSN: 16113349. DOI: 10.1007/978-3-319-93846-2_46.

[69] A. F. Gonçalves, A. M. Maciel, and R. L. Rodrigues, „Development of a data mining education framework for data visualization in distance learning environments“, *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, pp. 547–550, 2017, ISSN: 23259086. DOI: 10.18293/SEKE2017-130.

[70] D. Turnbull, R. Chugh, and J. Luck, „Learning management systems: a review of the research methodology literature in Australia and China“, *International Journal of Research and Method in Education*, vol. 44, no. 2, pp. 164–178, 2021, ISSN: 17437288. DOI: 10.1080/1743727X.2020.1737002.

[71] I. Dobre, „Learning Management Systems for Higher Education - An Overview of Available Options for Higher Education Organizations“, *Procedia - Social and Behavioral Sciences*, vol. 180, no. November 2014, pp. 313–320, 2015, ISSN: 18770428. DOI: 10.1016/j.sbspro.2015.02.122.

[72] T. I. Tawalbeh, „EFL Instructors' Perceptions of Blackboard Learning Management System (LMS) at University Level“, *English Language Teaching*, vol. 11, no. 1, p. 1, 2017, ISSN: 1916-4742. DOI: 10.5539/elt.v11n1p1.

[74] C. Hall, „Lighting a fire or filling a pail? Users' perceptions of a virtual learning environment“, *Survey Report, University of Swansea. Retrieved February*, vol. 28, p. 2008, 2006. [Online]. Available: https://cronfa.swan.ac.uk/Record/cronfa46045.

[75] H. Suleman, „Automatic marking with Sakai“, in *Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology*, ser. SAICSIT '08, Wilderness, South Africa: Association for Computing Machinery, 2008, 229–236, ISBN: 9781605582863. DOI: 10.1145/1456659.1456686.

[76] S. Dube and E. Scott, „An empirical study on the use of the Sakai Learning Management System (LMS): Case of NUST, Zimbabwe“, *In Proceedings of the e-skills for Knowledge Production and Innovation Conference*, no. November, pp. 101–107, 2014. DOI: 10.13140/RG.2.1.2589.1048.

[77] Z. Fariha, A. Zuriyati, and A. Kadir, „Comparing Moodle and eFront Software for Learning Management System“, *Australian Journal of Basic and Applied Sciences*, no. May, pp. 158–162, 2014, ISSN: 2309-8414. [Online]. Available: https://www.researchgate.net/profile/Aini-Kadir-2/publication/303137189_Comparing_Moodle_and_eFront_for_LMS/links/5737ef7d08ae9f741b2ad8e0/Comparing-Moodle-and-eFront-for-LMS.pdf.

[79] S. H. Gamage, J. R. Ayres, and M. B. Behrend, „A systematic review on trends in using Moodle for teaching and learning“, *International Journal of STEM Education*, vol. 9, no. 1, 2022, ISSN: 21967822. DOI: 10.1186/s40594-021-00323-x.

[80] B. C. Oguguo, F. A. Nannim, J. J. Agah, C. S. Ugwuanyi, C. U. Ene, and A. C. Nzeadibe, „Effect of learning management system on Student's performance in educational measurement and evaluation“, *Education and Information Technologies*, vol. 26, no. 2, pp. 1471–1483, 2021, ISSN: 15737608. DOI: 10.1007/s10639-020-10318-w.

[82] G. Snipes, „Product Review Google Data Studio“, *Journal of Librarianship and Scholarly Communication*, vol. 6, no. General Issue, pp. 0–5, 2018, ISSN: 2162-3309. DOI: 10.7710/2162-3309.2214.

[83] P. Michele, F. Fallucchi, and E. W. De Luca, „Create Dashboards and Data Story with the Data & Analytics Frameworks“, *Communications in Computer and Information Science*, vol. 1057 CCIS, no. April 2020, pp. 272–283, 2019, ISSN: 18650937. DOI: 10.1007/978-3-030-36599-8_24.

[85] C. T. Yang, T. Y. Chen, E. Kristiani, and S. F. Wu, „The implementation of data storage and analytics platform for big data lake of electricity usage with spark", *Journal of Supercomputing*, vol. 77, no. 6, pp. 5934–5959, 2021, ISSN: 15730484. DOI: 10.1007/s11227-020-03505-6.

[87] H. Snyder, „Literature review as a research methodology: An overview and guidelines", *Journal of Business Research*, vol. 104, no. March, pp. 333–339, 2019, ISSN: 01482963. DOI: 10.1016/j.jbusres.2019.07.039.

[88] P. Offermann, O. Levina, M. Schönherr, and U. Bub, „Outline of a design science research process", *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, DESRIST '09*, 2009. DOI: 10.1145/1555619.1555629.

[89] R. Baskerville, J. Pries-Heje, and J. Venable, „Soft design science methodology", *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, DESRIST '09*, no. January, 2009. DOI: 10.1145/1555619.1555631.

[90] I. Nurhas, S. Geisler, and J. M. Pawlowski, „Why Should the Q-Method Be Integrated Into the Design Science Research? a Systematic Mapping Study", *10th Scandinavian Conference on Information Systems, SCIS 2019*, 2019. [Online]. Available: https://aisel.aisnet.org/scis2019/9/.

[91] O. Sangupamba Mwilu, I. Comyn-Wattiau, and N. Prat, „Design science research contribution to business intelligence in the cloud — A systematic literature review", *Future Generation Computer Systems*, vol. 63, pp. 108–122, 2016, ISSN: 0167739X. DOI: 10.1016/j.future.2015.11.014.

[92] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, „Conceptual modeling for ETL processes", *ACM International Workshop on Data Warehousing and OLAP (DOLAP)*, pp. 14–21, 2002. DOI: 10.1145/583890.583893.

[94] Z. J. Liu, V. Levina, and Y. Frolova, „Information visualization in the educational process: Current trends", *International Journal of Emerging Technologies in Learning*, vol. 15, no. 13, pp. 49–62, 2020, ISSN: 18630383. DOI: 10.3991/ijet.v15i13.14671. [Online]. Available: https://www.learntechlib.org/p/217597/article_217597.pdf.

[96] R. Jin, G. Yang, and G. Agrawal, „Shared memory parallelization of data mining algorithms: Techniques, programming interface, and performance", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 71–89, 2005, ISSN: 10414347. DOI: 10.1109/TKDE.2005.18. [Online]. Available: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972726.5.

[97] P. Sedgwick, „Spearman's rank correlation coefficient", *BMJ (Online)*, vol. 349, no. August, 2014, ISSN: 17561833. DOI: 10.1136/bmj.g7327.

[98] B. Özmen and A. Altun, „Undergraduate Students' Experiences in Programming: Difficulties and Obstacles", *Turkish Online Journal of Qualitative Inquiry*, vol. 5, no. 3, p. 12, 2014. [Online]. Available: `https://dergipark.org.tr/tr/download/article-file/199844`.

[99] M. D. Siti Rosminah and M. A. Ahmad Zamzuri, „Difficulties in learning Programming: Views of students", *1st International Conference on Current Issues in Education (ICCIE2012)*, pp. 74–78, 2012. DOI: `10.13140/2.1.1055.7441`.

[100] E. Lahtinen, K. Ala-Mutka, and H. M. Järvinen, „A study of the difficulties of novice programmers", *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pp. 14–18, 2005. DOI: `10.1145/1067445.1067453`.

Book References

[16] F. Cady, *The data science handbook.* John Wiley & Sons, 2017, ISBN: 9781119092940.

[21] V. Ermolayev, F. Mallet, V. Yakovyna, H. C. Mayr, and A. Spivakovsky, *Correction to: Information and Communication Technologies in Education, Research, and Industrial Applications.* 2020, pp. C1–C1, ISBN: 9783319132051. DOI: `10.1007/978-3-030-39459-2_20`.

[22] S. Chacon and B. Straub, *Pro git.* Springer Nature, 2014, ISBN: 978-1-4302-1833-3.

[26] D. Westerveld, *API Testing and Development with Postman: A practical guide to creating, testing, and managing APIs for automated software testing.* Packt Publishing Ltd, 2021, ISBN: 978-1800569201.

[36] B Kovalerchuk, K Nazemi, R Andonie, N Datia, and ..., *Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery.* 2022, ISBN: 9783030931186.

[40] B. Mirkin, *Core data analysis: Summarization, correlation, and visualization.* Springer, 2019, ISBN: 978-3-030-00270-1. DOI: `https://doi.org/10.1007/978-3-030-00271-8`.

[46] C. M. Borror, *Statistical decision making.* ASQ Quality Press Milwaukee, WI, USA, 2009, pp. 418–472, ISBN: 978-0873897457.

[66] T. Soukup and I. Davidson, *Visual data mining: Techniques and tools for data visualization and mining.* John Wiley & Sons, 2002, ISBN: 978-0471149996.

[78] J. B. Idoko and J. Palmer, *A Comprehensive Review of Virtual E-Learning System Challenges.* Springer, 2023, pp. 141–151, ISBN: 978-3-031-42924-8. DOI: `https://doi.org/10.1007/978-3-031-42924-8_11`.

[81] D. G. Murray, *Tableau your data!: fast and easy visual analysis with tableau software.* John Wiley & Sons, 2013, ISBN: 978-1118612040.

[84] S. Shekhar, *Apache Superset Quick Start Guide: Develop interactive visualizations by creating user-friendly dashboards*. Packt Publishing Ltd, 2018, ISBN: 978-1788992244.

[86] A. Ferrari and M. Russo, *Introducing Microsoft Power BI*. Microsoft Press, 2016, ISBN: 9781509302758.

[93] P. Seamark, *Beginning DAX with Power BI: The SQL Pros Guide to Better Business Intelligence*. Apress, 2018, ISBN: 978-1484234761.

[95] H. Baars and H.-G. Kemper, *Datenbereitstellung und -modellierung*. 2021, pp. 15–90, ISBN: 9783834819581. DOI: 10.1007/978-3-8348-2344-1_2.

Online References

[2] A. Sherif. „Employment in the IT industry“. (2024), [Online]. Available: https://www.statista.com/topics/5275/employment-in-the-it-industry/ (visited on 09/18/2023).

[8] „Eductional Data Mining - Website“. (2023), [Online]. Available: https://educationaldatamining.org/ (visited on 09/23/2023).

[14] „TISS - Introduction in Programming 1“. (2023), [Online]. Available: https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=4035&dsrid=312&courseNr=185A91&semester=2023S (visited on 11/05/2023).

[15] „Who’s using Moodle?“ (2023), [Online]. Available: https://moodle.com/ (visited on 09/29/2023).

[17] „IntelliJ IDEA – the Leading Java and Kotlin IDE“. (2023), [Online]. Available: https://www.jetbrains.com/idea/ (visited on 12/10/2023).

[18] „What is Java technology and why do I need it?“ (2023), [Online]. Available: https://www.java.com/en/download/help/whatis_java.html (visited on 12/10/2023).

[19] L. S. Vailshery. „Most used programming languages among developers worldwide as of 2023“. (2023), [Online]. Available: https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/ (visited on 12/10/2023).

[24] K. Kelley. „What is GitLab?“ (2023), [Online]. Available: https://www.simplilearn.com/tutorials/git-tutorial/what-is-gitlab (visited on 12/15/2023).

[25] „Moodle - External Services“. (2023), [Online]. Available: https://moodledev.io/docs/apis/subsystems/external (visited on 12/20/2023).

[27] G. van Rossum. „Python Tutorial“. (2022), [Online]. Available: https://www.cse.unsw.edu.au/$\sim$en1811/python-docs/python-3.8.14-docs-pdf/tutorial.pdf (visited on 12/20/2023).

[28] „Jupyter Notebook: The Classic Notebook Interface“. (2023), [Online]. Available: https://jupyter.org/ (visited on 12/20/2023).

[30] „Tutorial: Fabric for Power BI users“. (2024), [Online]. Available: https://learn.microsoft.com/en-us/power-bi/fundamentals/fabric-get-started (visited on 02/02/2023).

[35] T. L. Edu. „Grading Systems Around the World“. (2023), [Online]. Available: https://leverageedu.com/blog/grading-systems/ (visited on 02/22/2024).

[43] „ORDINARY LEAST SQUARES REGRESSION (OLS)“. (2024), [Online]. Available: https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols (visited on 03/05/2024).

[47] „Spearman’s Rank Correlation Coefficient Rs and Probability (p) Value Calculator“. (2024), [Online]. Available: https://geographyfieldwork.com/SpearmansRankCalculator (visited on 03/05/2024).

[73] „Blackboard - Building Blocks and REST APIs“. (2024), [Online]. Available: https://help.blackboard.com/de-de/Learn/Administrator/SaaS/Integrations/Compare_Building_Blocks_and_Rest (visited on 01/02/2024).