From Collecting, Integrating, and Visualizing Student Data to Predicting Student Dropout and Performance

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Alexander Askinadze

aus Winniza

Düsseldorf, Mai 2020

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Stefan Conrad

2. Prof. Dr. Stefan Harmeling

Tag der mündlichen Prüfung: 05.10.2020

Ich versichere an Eides Statt, dass die Dissertation von mir selbstständig und ohne unzulässige fremde Hilfe unter Beachtung der *Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf* erstellt worden ist.

Die hier vorgelegte Dissertation habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

Düsseldorf, den 05.05.2020 Alexander Askinadze

Dedicated to my parents

Acknowledgements

First and foremost, I would like to express my gratitude to my thesis advisor, Prof. Dr. Stefan Conrad, for the opportunity to conduct my research in his group and for his supervision and motivation throughout my studies. Additionally, I thank Prof. Dr. Stefan Harmeling for agreeing to mentor me and for being the second reviewer.

I thank my colleagues Dr. Matthias Liebeck and Dr. Pashutan Modaresi for the cooperation, support and the opportunity to learn a lot from them.

Furthermore, I thank my colleagues Kirill Bogomasov, Daniel Braun, Janine Golov, Dr. Ludmila Himmelspach, Gerhard Klassen, Martha Krakowski, Magdalena Rischka, Julia Romberg, and Dr. Michael Singhof for the fruitful collaboration and interesting scientific discussions.

In addition, I would like to thank Guido Königstein and Sabine Freese for their excellent technical and administrative support, which enabled us to have an optimal working environment.

I would like to express gratitude to IST University of Applied Sciences, especially Dr. Hans E. Ulrich, Dr. Katrin Gessner-Ulrich, Marco Gensmüller, and Martin Sommer who supported my research.

I am also grateful to istis Informationssysteme and Frank Beilke for their support and cooperation during the research period.

I also want to thank my friends, parents and my partner Stephanie for their support. Without you it would not have been possible.

Abstract

Advancing digitization is leading to more and more data, making the storage, integration, and processing of these large amounts of data (often referred to as big data) a challenge. As digitization extends to many areas of life, it also changes education immensely, so that education increasingly takes place in digital learning environments. Several research areas, such as educational data mining and learning analytics, have been established, which deal with the collection and analysis of student data from various perspectives. In this thesis, we deal with the following question: **How can educational data be collected, integrated, analyzed, visualized, and finally used to predict dropout and performance?**

To answer this question, we described the development of a dashboard in which we integrated data from hierarchical and non-hierarchical modules of two study programs. We proposed visualization techniques based on Sankey, UpSet, and Venn diagrams, which we used to visualize student progress for different cohorts. We found that a small amount of student data (information whether exams have been passed or not) is sufficient to predict student dropout and proposed a new method that uses the temporal aspect of student progress data to predict dropout.

If digital learning is offered in learning management systems (LMS) in addition to the usual teaching, then students, for example, open teaching materials in the LMS or learn with interactive learning elements offered by third party providers. This results in heterogeneous data that can be analyzed. Therefore, we investigated how the student interaction data from different digital learning environments can be tracked and integrated in order to apply data mining to this data. Based on several case studies, we investigated how interaction data can be used to predict student performance.

Since the application of learning analytics and educational data mining is not possible without considering data protection issues, we investigated what universities have to consider if they want to conduct learning analytics and educational data mining in compliance with GDPR. We found that educational institutions must be open and transparent in providing information about what data is stored and for what purposes it is processed. If the purposes of processing are well-argued, student consent is not always necessary. However, if students have the choice of which of their data may be tracked and stored, this leads to data sets with missing values. Therefore, we investigated the extent to which the missing values can be predicted from existing data in order to create better predictive models.

Zusammenfassung

Die fortschreitende Digitalisierung führt zu immer mehr Daten, was die Speicherung, Integration und Verarbeitung dieser großen Datenmengen (oft als Big Data bezeichnet) zu einer Herausforderung macht. Da sich die Digitalisierung auf viele Lebensbereiche ausdehnt, verändert sie auch die Bildung immens, so dass Bildung zunehmend in digitalen Lernumgebungen stattfindet. Es wurden mehrere Forschungsbereiche, wie z.B. Educational Data Mining und Learning Analytics eingerichtet, die sich mit der Sammlung und Analyse von studentischen Daten aus verschiedenen Perspektiven befassen. In dieser Arbeit beschäftigen wir uns mit der folgenden Frage: **Wie können Bildungsdaten gesammelt, integriert, analysiert, visualisiert und schließlich zur Vorhersage von Studienabbruch und studentischen Leistungen verwendet werden?**

Um diese Frage zu beantworten, haben wir die Entwicklung eines Dashboards beschrieben und verschiedene Visualisierungstechniken zur Visualisierung des Studentenfortschritts für verschiedene Kohorten vorgeschlagen, die auf Sankey-, UpSet- und Venn-Diagrammen basieren. Wir fanden heraus, dass eine kleine Menge an Daten der Studierenden (Informationen darüber, ob Prüfungen bestanden wurden oder nicht) ausreicht, um den Studienabbruch vorherzusagen. Dazu haben wir eine Methode vorgeschlagen, die den zeitlichen Aspekt der Studienverlaufsdaten zur Vorhersage nutzt.

Wenn digitales Lernen in Lernmanagementsystemen (LMS) zusätzlich zum üblichen Unterricht angeboten wird, dann öffnen die Studierenden beispielsweise Lehrmaterialien im LMS oder lernen mit interaktiven Lernelementen, die von Drittanbietern angeboten werden. Dies führt zu heterogenen Daten, die analysiert werden können. Daher haben wir untersucht, wie die aus Interaktionen mit digitalen Lernelementen resultierenden Daten gesammelt und integriert werden können. Auf der Grundlage mehrerer Fallstudien haben wir untersucht, wie Interaktions-Daten zur Vorhersage von studentischen Leistungen verwendet werden können.

Da die Anwendung von Learning Analytics und Educational Data Mining ohne Berücksichtigung von datenschutzrechtlichen Fragen nicht möglich ist, haben wir untersucht, was Bildungseinrichtungen beachten müssen, wenn sie studentische Daten DSGVO konform nutzen wollen. Dabei haben wir festgestellt, dass Bildungseinrichtungen offen und transparent Auskunft darüber geben müssen, welche Daten gespeichert und zu welchen Zwecken sie verarbeitet werden. Wenn die Zwecke der Verarbeitung gut begründet sind, ist die Zustimmung der Studierenden nicht immer erforderlich. Wenn Studenten jedoch die Wahl gegeben wird, welche Daten die Bildungseinrichtungen tracken und speichern dürfen, dann führt dies zu Datensätzen mit fehlenden Werten. Wir haben untersucht, inwieweit sich die fehlenden Werte aus den vorhandenen Daten vorhersagen lassen, um bessere Vorhersagemodelle zu erstellen.

Contents

1 Introduction

"Information is the oil of the 21st century, and analytics is the combustion engine."
— Peter Sondergaard (Pettey, 2011)

We are in the midst of a technological revolution, a time in which substantial technological progress is leading to paradigm shifts, also called industrial revolutions. After mechanization (1st industrial revolution), mass production thanks to electricity and assembly line (2nd industrial revolution), and automation through computer technologies (3rd industrial revolution), we are now in the age of smart and networked systems. This advancing digitization is often referred to as the 4th industrial revolution (Lasi et al., 2014). This digitization is leading to more and more data, making the storage, integration, and processing of these large amounts of data (often referred to as big data) a challenge. As digitization extends to many areas of life, it also changes education immensely, so that education increasingly takes place in digital learning environments. In this thesis, we deal with the following question:

> How can educational data be collected, integrated, analyzed, visualized, and finally used to predict dropout and performance?

The remainder of this chapter is structured as follows: In Chapter 1.1, we introduce the research field of educational data mining. Then, in Chapter 1.2, we describe the detailed research questions this work deals with and in Chapter 1.3, we show our contribution to this research field. Finally, in Chapter 1.4, we give an overview of the remaining chapters of this thesis.

1.1 Educational Data Mining

Several research areas have been established, which deal with the collection and analysis of student data from various perspectives. A large research community deals with *Educational Data Mining* (EDM). In this research field, methods are developed that are particularly suitable for data mining on data from educational contexts. A similar field of research is *Learning Analytics* (LA), which also deals with data mining methods, but is primarily concerned with the collection, analysis, and reporting of data from educational contexts. Romero and Ventura (2013) visualized EDM as the intersection of the three disciplines Computer Science, Statistics, and Education and LA as the intersection of Education and Statistics. Siemens and Baker (2012) provided a detailed analysis of the similarities and differences between the two research communities.

Romero and Ventura (2013) stated that there are several international societies dealing with EDM and LA. The most important are the International *Educational Data Mining Society*[1] and the *Society for Learning Analytics Research* (SoLAR)[2]. The most relevant conferences in this field are the *Educational Data Mining Conference* and the *Learing Analytics & Knowledge (LAK) Conference*.

The first and most important literature reviews in the EDM area include Romero and Ventura (2007) and Baker and Yacef (2009). Important updates were made in Romero and Ventura (2010) and Romero and Ventura (2013).

Romero and Ventura (2013) cited 67 previous studies and summarized the main goals and methods in EDM research. The authors mentioned that the goals that are pursued with the use of EDM can be very different and depend heavily on the user type. Potential users or stakeholders are students, educators (teachers, instructors, and tutors), researchers, and administrators. They also mentioned that there are a variety of potential problems and objectives for each type of stakeholder and listed the following topics of interest in the EDM research: developing methods for EDM, mining educational interaction data, data-driven adaptation in educational environments, improving educational software, improving teacher support, and many more. Furthermore, the authors listed popular DM methods, which are also used in EDM for problem-solving. These include the typical DM methods such as prediction, clustering, outlier detection, relationship mining, social network analysis, process mining, text mining, distillation of data for human judgment, as well as special EDM methods such as *knowledge tracing* (KT) (Corbett and Anderson, 1994). The authors gave the following examples of tasks in educational environments that can be solved using EDM methods: predicting student performance, providing feedback for supporting instructors, recommendations to students, student modeling, profiling students, constructing courseware, and many more.

In a recent detailed review, Aldowah et al. (2019) mentioned that the previous reviews have provided significant insight into this fast-growing field and, at the same time, criticized that there is little evidence on the association between educational problems and EDM (or LA) methods that can solve these problems. In their own review, they selected 402 articles from a total of 1200 originally collected articles from the years 2000-2017, which they considered to be the most valuable due to some quality criteria. They categorized the EDM applications in the examined articles into four topics, which we briefly summarize:

- **Learning analytics**: Instructors or tutors need methods to measure the effectiveness of a course and initiate possible interventions. DM methods are used to derive actionable information based on students' interactions in a digital learning environment. Widely used techniques are classification, statistics, clustering, and visual data mining (Romero and Ventura, 2007).

[1] `http://www.educationaldatamining.org`

[2] `http://www.solareaearch.org`

- **Predictive analytics**: This is the topic to which most of EDM articles belong (63%). In the context of EDM, predictive analytics is about predicting student performance and student dropout based on different student data such as exam results, participation, engagement, grades, and domain knowledge. For this topic, classification and clustering are the most commonly used (Romero and Ventura, 2013).

- **Behavioral analytics**: In this topic, researchers use DM techniques to analyze student behaviors in digital learning environments to better understand students and improve their learning experience. Clustering, classification, association rule mining, and visualization are used most frequently to achieve the goals (Romero and Ventura, 2007).

- **Visualization analytics**: The goal of this topic is visualizing complex student data, e.g., the visualization of student interactions in digital learning environments (Romero and Ventura, 2007).

1.2 Current Research Questions

Romero and Ventura (2007) visualized the application of DM to educational systems as an iterative process. Figure 1.1 shows a modified and extended version based on their visualization. The bold arrows show the path from a student to *institutional administrators* (defined by Müller (1985) as "executives who are responsible for the organization of the purposeful overall functioning of institutions of higher education"). In the following, we identify different research questions that are based on the steps along the drawn path.

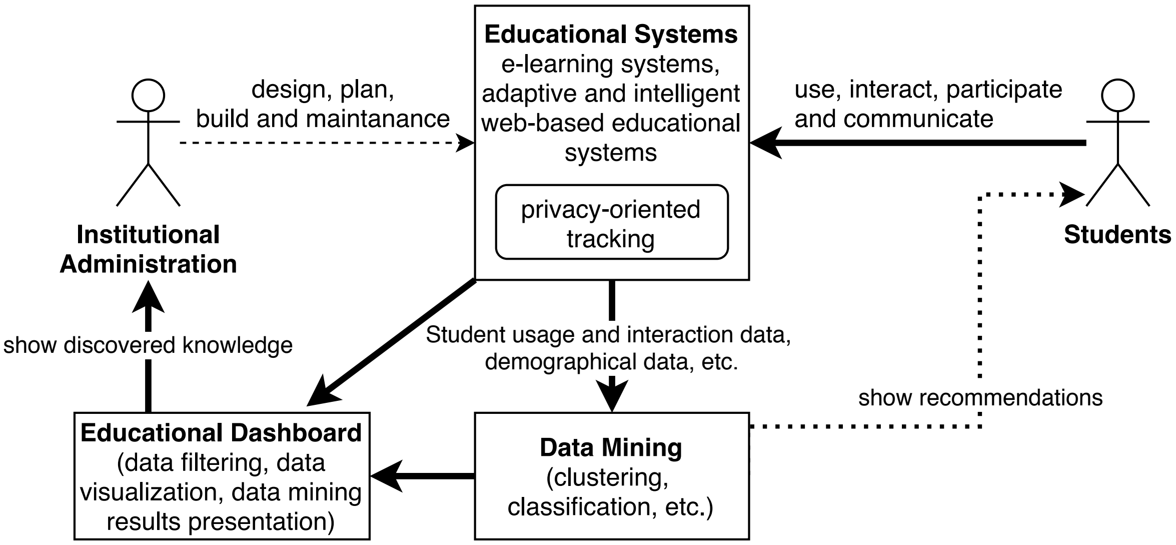

Educational Systems
e-learning systems, adaptive and intelligent web-based educational systems
privacy-oriented tracking
design, plan, build and maintanance
use, interact, participate and communicate
Institutional Administration
Students
show discovered knowledge
Student usage and interaction data, demographical data, etc.
show recommendations
Educational Dashboard (data filtering, data visualization, data mining results presentation)
Data Mining (clustering, classification, etc.)

Figure 1.1: The cycle of applying data mining in educational systems. This is an extended version of the cycle presented by Romero and Ventura (2007).

The questions examined can be divided into the following main categories: collecting and integrating educational data (Chapter 1.2.1), data privacy (Chapter 1.2.4), predictive analytics (Chapter 1.2.2), and visualization analytics (Chapter 1.2.3). We number the questions and show in Chapter 1.3, how we contribute to these questions.

1.2.1 Collecting and Integrating

In his keynote to the Learning Analytics & Knowledge Conference 2018, Ryan Baker (the founding president of the International Educational Data Mining Society) named the most important unsolved problems in the research areas EDM & LA. Among other things, he named the problem of the *learning system wall*: Individual learning systems are already able to learn a lot from the data of a student, but as soon as a student uses a different learning system, the new system has to start from scratch to train the models for this student.

The solution to this problem is not trivial because the data available in the EDM area can be very heterogeneous. Romero and Ventura (2013) described that different levels of granularity (from coarse to fine grain) or different hierarchy levels, such as clickstream data, student data, and course data, can be given. Therefore, smart data integration strategies and extensive preprocessing to apply data mining are required (Daniel, 2015).

We want to develop a dashboard that is able to integrate administrative student data from different German universities. Although the data of the different universities are similar, there are some differences, e.g., the modules (and exams) of their study programs and courses could be flat (not hierarchical) or hierarchically structured. Therefore, we address the following question:

RQ1: How can student progress data from different universities be integrated into a single (dashboard) system in order to avoid the development of separate (dashboards) systems for each educational institution?

We also consider other data sources that may contain educational data, including services that offer digital learning elements. Various studies have dealt with the issue of how widespread and popular digital learning elements are. For example, in the *Student Watch study* (National Association of College Stores, 2018), which examines students from various North American education institutions, it is reported that 25% of students who have purchased at least one course material also bought a digital version. Two years earlier, the percentage was 10% smaller. A recent German study (Rat für kulturelle Bildung, 2019) has shown that about 50% of German students between the ages of 12 and 19 are learning with YouTube. In another German study, Bialecki (2013) examined children under the age of 13 and showed that the children in this cohort prefer to learn with computers, tablets, or smartphones. This development shows that the demand for digital learning media will continue to increase over the next few years. Bishka and Fedy (2018) mentioned that the generation of digital natives likes to learn with electronic devices and videos, but print versions continue to be popular. Digital learning materials do not seem to replace print media but are becoming increasingly popular as an add-on. Therefore, we address the following question:

RQ2: Modern learning environments offer a variety of possible new learning elements, such as interactive quizzes and videos. These can be offered on different devices and by different third-party providers. How can the student interaction data be tracked and integrated in order to apply data mining to this data?

1.2.2 Predictive Analytics

In this thesis, we deal with the following two tasks in the field of predictive analytics: prediction of student dropout in higher education and prediction of student performance.

In recent years, various studies such as Dekker et al. (2009), Manrique et al. (2019), Hartl (2019), Berens et al. (2019), and Aulck et al. (2019) examined the dropout prediction in higher education. The machine learning methods used for classification are mostly similar. These include neural networks, support vector machines, decision trees, and random forest (Alban and Mauricio, 2019b). The studies also have in common that students are modeled by their demographic data and previous exam performance. Most of them conclude that demographic performance does not have as much predictive power as previous academical performance. They mostly use only the latest known state of academical performance to model the feature vectors. This means, for example, that for a student in the third semester, it is not taken into account how the student has studied until the end of the third semester (e.g., how the exam results were in the meantime), but only what the student has achieved by the end of the third semester. The studies differ in the way they created feature vectors. This leads us to the following research questions:

RQ3: How can student data in higher education be used to predict student dropout? We consider this question under the following aspects:

- What data is sufficient to predict student dropout?
- How should the feature vectors be coded?
- How can the temporal information of student progress be used to improve the predictions?

Unlike dropout prediction, student performance prediction is most often about predictions at course level. Various papers have investigated the prediction of final course grades or grade levels (Al-Radaideh et al., 2006; Cortez and Silva, 2008; Shahiri et al., 2015; Saa et al., 2019). Saa et al. (2019) found that demographic data, performance data, and behavioral e-learning data are the most important data for predicting student performance. If educational institutions provide students with the materials in learning management systems (LMS), the behavioral LMS data (describing students use an LMS) can be used to predict student performance. Al-Radaideh et al. (2006) examined the combination of demographic and behavioral LMS usage data and found that this combination can be used to predict student performance. However, it is not clear how much the LMS behavioral data alone contributes to the prediction. The number of demographical features available to model the students can be very large. To simplify the model, it may make sense to use only those student features that are relevant for the prediction. This leads us to the following research question:

RQ4: How well can student performance be predicted based on demographical and behavioral LMS usage data? We consider this question under the following additional aspects:

- How well is the prediction if only behavioral LMS usage data is given?
- How to find a small feature set sufficient to predict the student performance?

As already written in the motivation to question *RQ*2, there are more and more digital learning materials, which enable students to learn interactively, e.g., answering quiz questions in video lectures[3]. An example of a digital teaching material delivery system is BookRoll (Flanagan and Ogata, 2017), a digital eBook reader with behavior sensors, which enable to track usage behavior (e.g., clicking, zooming, and browsing). Such systems are usually offered by third-party providers and, therefore, learning takes place outside the actual LMS. Interactions produced by these systems can be tracked and, therefore, produce large amounts of data that can be analyzed using DM techniques. This leads to the following research question:

RQ5: How can the exam grades be predicted based on usage behavior in digital learning, e.g., an eBook?

1.2.3 Visualization Analytics

Schwendimann et al. (2016) examined 55 selected articles on educational dashboards. They found that only one dashboard was intended for administrative monitoring (most existing dashboards are intended for teachers and students). We have developed a dashboard for administrative monitoring and considered the following question:

RQ6: How can the study progress data of students in higher education be visualized? We consider this question under the following aspects:

- How to visualize which exams or exam combinations graduates or dropout students passed until a selected semester?
- How to identify exams that are difficult for students using visualizations?
- How to compare the study progress of different cohorts of students in one visualization?

1.2.4 Data Privacy

Another topic that has become important in recent years is the handling of data with regard to data privacy. Data privacy is an important issue whose omission allows no practical use of EDM and LA. The *General Data Protection Regulation* (GDPR) makes many requirements for the handling of student data. In literature, there are several discussions on how to handle data privacy in learning analytics. Often it is about giving students the opportunity to decide for themselves what is stored about them and that the data will not be passed on to third parties (Trainor, 2015; El-Khattabi, 2017). Few studies have presented frameworks and architectures that deal with current legal requirements (Cormack, 2016). This leads to the following research questions:

RQ7: What do universities have to consider if they want to conduct learning analytics and educational data mining in compliance with GDPR?

RQ8: If students can choose which of their data may be processed for specific purposes, data sets with missing values will be created. Can such data sets still be used to train useful predictive models after applying missing value imputation strategies?

[3] Services for creating video learning elements with quizzes: `www.lernblitz.de`, `www.h5p.com`

1.3 Contributions

This thesis consists of a mixture of new content (only included in this thesis) and summarized contents based on papers published at international and peer-reviewed conferences and workshops. In this chapter, we show which chapters and papers deal with the research questions raised in the previous sections. Therefore, in Table 1.1, we present a mapping of our research questions to the relevant chapters, papers, and contributions.

Research question	Research topic	Chapter	Publication
RQ1	Intergrating	Chapter 4.1	Askinadze and Conrad (2018a)
RQ2	Tracking & Collecting & Intergrating	Chapter 3.3	Askinadze and Conrad (2017)
RQ3	Predictive analytics	Chapter 3.4	Askinadze and Conrad (2019)
RQ4	Predictive analytics	Chapter 3.5.1	
RQ5	Predictive analytics	Chapter 3.5.2	Askinadze et al. (2018) Askinadze et al. (2019a)
RQ6	Visualization analytics	Chapter 4.1.2.1	Askinadze and Conrad (2018a) Askinadze et al. (2019b)
RQ7	Data privacy	Chapter 4.3	
RQ8	Data privacy	Chapter 3.5.1	Askinadze and Conrad (2018b)

Table 1.1: Research questions and contributions

In addition, the following publications in the field of data science have been published that do not directly contribute to the research questions:

- In Askinadze (2016), we conducted a brief review of the application of support vector regression (SVR) to predict student grades and evaluated the SVR approach with different parameters on a public student data set.
- In Liebeck et al. (2016), we examined how we can predict “the big five” personality traits of students using their program code.
- In Hirmer et al. (2017), we described our approach to solving the *BTW2017 Data Science Challenge* in which we achieved the second place.

1.4 Outline of this Thesis

This thesis is structured as follows. In Chapter 2, we give an overview of machine learning and show the theoretical background of the classifiers and the definitions of the evaluation measures we need for predicting student dropout and precision. In Chapter 3, we show the steps necessary to predict student performance and dropout, and evaluate the proposed approaches using different data sets. In Chapter 4, we present the development of our administrative educational dashboard and finally draw the conclusion in Chapter 5.

2

Machine Learning

> *"A computer program is said to* ***learn*** *from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*
>
> — Tom Mitchell (Mitchell, 1997)

Data mining (DM) is a field of research that, among other things, uses machine learning techniques to extract new and meaningful knowledge from databases (Mitchell, 1997). Almost all machine learning methods are also used in the EDM area. Most of the ML methods used in this context are differentiated into *supervised* and *unsupervised* methods.

The point of supervised methods is to train an ML algorithm $f : X \rightarrow Y$ based on a training set $D = \{(x_i, y_i), x_i \in X, y_i \in Y\}$ where f learns a mapping function between X and Y, so that $f(x_i) = y_i$. The goal of training is *generalization*, which means that after the training the mapping will also work for unseen tuples $(x^*, y^*) \notin D$, such that $f(x^*) = y^*$ applies. The assignment of a value y^* to an unseen input vector x^* is called prediction. How well the ML algorithms can generalize depends not only on the idea of the algorithm itself but also on the given training data, which often represents only a small part of the possible input data.

If the values of y_i are categorical (belong to a finite set of categories), then f solves a *classification* problem. The classes are often represented by a set $\{z_i \in \mathbb{Z}\}_{1 \leq i \leq C}$, where C is the number of classes. For $C = 2$, we speak of a binary classification (e.g. student dropout prediction with the two classes "dropout" and "graduate"). For y_i, we write $y_i \in \{0, 1\}$ or $y_i \in \{-1, 1\}$, where one class is associated with 1 and the other with 0 or -1. For $C > 2$ (if for example several semantic classes are used, which describe the level of knowledge of students like "low knowledge", "middle knowledge", and "high knowledge"), it is called *multi-class classification* and for y_i then $y_i \in \{1, ..., C\}$ applies. If $y_i \in \mathbb{R}^m$ are continuous variables, f solves a *regression* problem (Bishop, 2006; Murphy, 2012).

With unsupervised methods, there are no labels y_i, so the training set is simply a set of multidimensional vectors $D = \{x_i \in \mathbb{R}^n\}$. The goal is not to learn a mapping function, but rather to find interesting patterns in the data. This process is sometimes called *knowledge discovery*. One of the most popular unsupervised ML procedures is *clustering*, which can find groups of similar input data (Bishop, 2006; Murphy, 2012).

A further distinction of the ML algorithms is given by the terms *instance-based learning* and *model-based learning*. In instance-based learning, the training data is stored (wholly or partially), as it is necessary to classify new data. In model-based learning, a model is learned on the training data, once all model parameters have been learned, the training data is no longer needed to classify new data (Géron, 2018).

Some of the most popular supervised ML procedures are presented in Chapter 2.1. In order to measure how well the supervised ML methods are able to generalize, evaluation measures are needed, which are presented in Chapter 2.2.

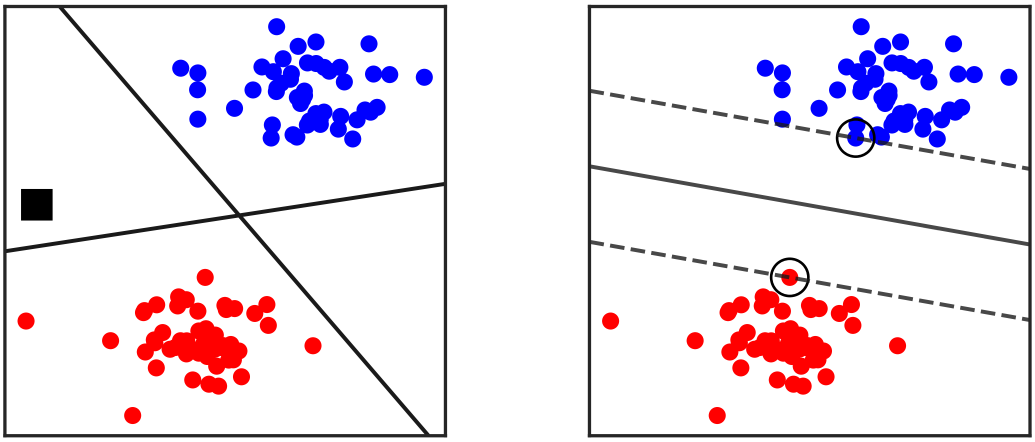
2.1 Supervised Learning

Most popular and frequently-used ML methods for the student dropout prediction task include the *support vector machine*, *decision tree*, *random forest*, and *neural networks* (Alban and Mauricio, 2019b), which are presented in this chapter.

2.1.1 Support Vector Machine

In this chapter, we show the basics of the *support vector machine* (SVM) (Cortes and Vapnik, 1995). Our explanations on SVM are based on Cortes and Vapnik (1995), Bennett and Campbell (2000), Manning et al. (2008), and Webb and Copsey (2011).

The SVM is a supervised binary classifier $f : \mathbb{R}^n \rightarrow \{-1, +1\}$, which finds a separation plane for two linearly separable sets of points. Inspired by VanderPlas (2016), we visualize the idea behind the SVM using several examples. Figure 2.1a shows two linearly separable 2D point sets (red and blue). There are infinitely many possible hyperplanes to separate these two linearly separable sets of points from each other.



(a) Several possible separation planes

(b) Optimal separation plane

Figure 2.1: Possible and optimal separating planes in the separable case

Let $x^* \in \mathbb{R}^n$ be a new point (represented as a black square in Figure 2.1a). The decision regarding which of the two classes $\{+1, -1\}$ (represented as blue or red) x^* should be assigned can be made depending on which side of the separation plane the point is located. As shown in the example, this decision would depend on which of the two separating planes is used. This motivates finding the optimal separating plane. The task of training an SVM is to find the optimal separating plane so that the margin between classes is maximized.

Let $w \in \mathbb{R}^n$ be a n-dimensional vector and $b \in \mathbb{R}$ a scalar. The classification rule for a point x^* where $w^T x^*$ is the dot product can be specified by:

$$f(x^*) = \text{signum}(w^T x^* + b) = \begin{cases} +1, & \text{if } w^T x^* + b \geq 0 \\ -1, & \text{if } w^T x^* + b < 0 \end{cases} \tag{2.1}$$

To find the optimal parameters w and b, the following optimization problem based on training data given by $\{(x_1, y_1), ..., (x_L, y_L) | x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}\}$, where x_i are the training points and y_i the corresponding classes, must be solved:

$$\min \frac{||w||^2}{2} \text{ s.t. } \quad y_i(w^T x_i + b) \geq 1 \ \forall i \in \{1, .., L\} \tag{2.2}$$

The corresponding Lagrangian function is given by:

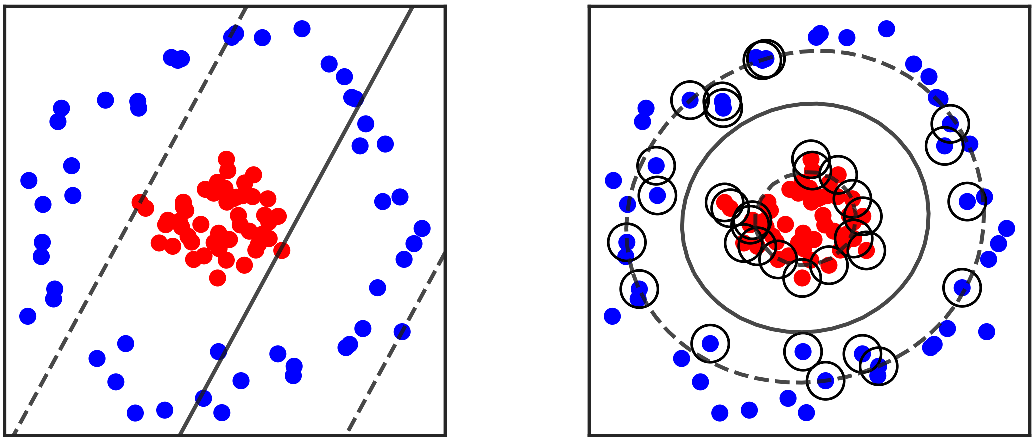
$$\mathcal{L}(w, b, \alpha) = \frac{||w||^2}{2} - \sum_i \alpha_i [y_i(w^T x_i + b) - 1] = \frac{||w||^2}{2} - \sum_i \alpha_i y_i w^T x_i - b \sum_i \alpha_i y_i + \sum_i \alpha_i$$

For an optimal w and b, $\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial w} = w - \sum_i \alpha_i y_i x_i = 0$ and $\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial b} = -\sum_i \alpha_i y_i = 0$ applies, so that we derive $w = \sum_i \alpha_i y_i x_i$ and $\sum_i \alpha_i y_i = 0$. After substituting w and simplifying the Lagrangian we derive the dual optimization problem:

$$\max \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad \text{s.t.} \quad \alpha_i \geq 0, \sum_{i=0}^{L} \alpha_i y_i = 0 \tag{2.3}$$

This dual representation can be formulated as a *quadratic programming* (QP) problem with linear constraints, which can be solved efficiently using *sequential minimal optimization* (Platt, 1998). Usually only a small subset of the training points (x_i where $\alpha_i > 0$) determines the exact position of the separation plane (Manning et al., 2008). These points, which are closest to the hyperplane, are called *support vectors*. Since the SVM is based on a subset of training data, it can be considered an instance-based learning algorithm (Domingos, 2012). In Figure 2.1b, the two support vectors are marked by a black circle.

Figure 2.3a shows two non-linear separable 2d point sets (red and blue), where the red point set is completely enclosed by the blue one. In Figure 2.2a, we have drawn the separating plane found by the linear SVM, which is obviously not optimal. If the training set cannot be separated linearly, a function $\phi(x) : \mathbb{R}^n \to \mathbb{R}^d$ could be found which transforms a n-dimensional vector $x \in \mathbb{R}^n$ into a d-dimensional vector $\phi(x) \in \mathbb{R}^d$ with $\phi(x) = (\phi_1(x), ..., \phi_d(x))$. With a suitable function $\phi(x)$, the data in the higher dimensional space could be separated linearly with higher probability. The function $\phi(x)$



(a) Attempt to separate non-linearly separable sets of points using linear kernel.

(b) Decision boundary created using RBF kernel

Figure 2.2: Decision boundaries created by linear and RBF SVM.

is called *feature map*, that transforms the data into the *feature space* (Hofmann et al., 2008). The example in Figure 2.3b shows the transformation of the two-dimensional sets of points into a three-dimensional space in which a separation hyperplane can be found.

For optimal w and b, the rule of equation 2.1 can be reformulated to equation 2.4:

$$f(x^*) = \text{signum}(w^T\phi(x^*) + b) = \begin{cases} +1, & \text{if } w^T\phi(x^*) + b \geq 0 \\ -1, & \text{if } w^T\phi(x^*) + b < 0 \end{cases} \tag{2.4}$$

For an optimal w, $\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial w} = w - \sum_i \alpha_i y_i \phi(x_i) = 0$ applies, so that we get $w = \sum_i \alpha_i y_i \phi(x_i)$ and the product $w^T\phi(x^*)$ from equation 2.4 can be specified by equation 2.5:

$$w^T\phi(x^*) = \phi(x^*)^T w = \phi(x^*)^T \sum_{i=1}^{L} \alpha_i y_i \phi(x_i) = \sum_{i=1}^{L} \alpha_i y_i \phi(x^*)^T \phi(x_i) \tag{2.5}$$

The scalar product $\phi(x^*)^T\phi(x_i)$ in equation 2.5 can be replaced by a function K:

$$K(x, y) = \phi(x)^T\phi(y) \tag{2.6}$$

If we know function $K(x, y)$, we do not need to calculate $\phi(x)$ and $\phi(y)$ explicitly. This concept is called kernel trick and the classification rule for a point x^* using a kernel function is then specified as follows:

$$f(x^*) = \text{signum}\left(\sum_{i=1}^{L} \alpha_i y_i K(x^*, x_i) + b\right) \tag{2.7}$$

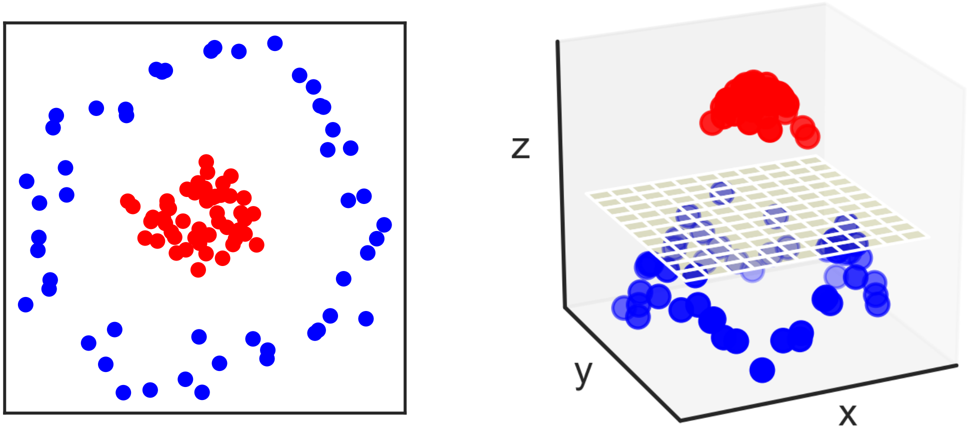
There are many different kernels besides the *linear kernel* $K(x, y) = x^T y$. One of the most popular kernels is the *radial basis function* (RBF) kernel, which is defined by

$$K_{RBF}(x, y) = \exp(-\gamma||x - y||^2) \tag{2.8}$$

The RBF kernel is often referred to in the literature as the Gaussian kernel. In the guide (Hsu et al., 2003) to the libSVM library (Chang and Lin, 2011), it is mentioned that the RBF kernel is a useful first choice. In Figure 2.2b, we can see from the decision boundary that the RBF kernel is able to separate the two sets of points from each other, unlike the linear kernel in Figure 2.2a. The required support vectors are marked by circles.

However, the use of kernels does not guarantee that the data in the feature space can be linearly separated. The solution is to use the *soft-margin* SVM. The constraint $y_i(w^T x^* + b) \geq 1$ is updated to $y_i(w^T x^* + b) \geq 1 - \xi_i$, so that ξ_i can be used to control which points may violate the constraints. The target function $\frac{||w||^2}{2}$ (equation 2.2) is extended by $C(\sum_i \xi_i)$, so that the optimization problem in equation 2.9 is given (Cortes and Vapnik, 1995). Useful hyperparameters C and γ for an SVM with the RBF kernel can be found with a grid search (Hsu et al., 2003).

$$\min \frac{||w||^2}{2} + C(\sum_i \xi_i) \text{ s.t. } \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \; \forall i \in \{1, .., L\} \tag{2.9}$$



(a) Not linear separable point sets. (b) Transformation into a higher dim. space.

Figure 2.3: SVM kernel trick

For multi-class problems, there are different approaches to the use of SVM, such as one-vs-one, one-vs-all (Hsu and Lin, 2002), or the *direct acyclic graph* (DAG) SVM (Platt et al., 2000).

The idea of the SVM can also be used for regression, which is called *support vector regression* (SVR) (Drucker et al., 1997; Smola and Schölkopf, 2004).

2.1.2 Tree Structured Classifiers

Alban and Mauricio (2019b) and Agrusti et al. (2019) state that decision trees (DT) and random forests are the most commonly-used classifiers for the task of dropout prediction according to literature. In this chapter, we explain both approaches.

2.1.2.1 Decison Tree

A decision tree is a hierarchical structure, more precisely a directed acyclic rooted tree. The training of a decision tree begins with a root node, whereby the training set is recursively divided into several splits/child nodes. The division of the training data into child nodes is performed based on an attribute/feature in the data. The selection of the attribute in each node is decided on the basis of a measure such as *information gain* or *gini impurity*. The splitting procedure is executed until an abort criterion, resulting in leaf nodes. Each leaf node is associated with a class label, which can be selected based on the majority class of the elements in the leaf node. If a tree has been trained, and a new instance x should be classified, then x follows the path of satisfied conditions from the root to a leaf node and is classified with the class label of the reached leaf node (Agrusti et al., 2019; Nandeshwar et al., 2011; Hu et al., 2014; Romero et al., 2008).

Agrusti et al. (2019) state in their systematic review that the decision trees *Iternative Dichotomizer* (ID3) (Quinlan, 1986), C4.5 (Quinlan, 1993), C5 (Quinlan, 2018) and ***C**lassification **A**nd **R**egression **T**rees* (CART) (L. Breiman et al., 1984) are among the most commonly used for the student dropout prediction task. The differences between the individual decision trees are as follows:

- C4.5 is an extension of the ID3 algorithm. While ID3 can only deal with discrete variables, C4.5 can also deal with continuous variables (a threshold is calculated for the split attribute and the training data is divided among the children nodes depending on whether the corresponding attribute value is above or below the threshold). In addition, the data may have missing values in contrast to ID3. Moreover, pruning methods (branches that are not helpful are removed after the tree is finished) are applied to the resulting tree (Nandeshwar et al., 2011; R. Pandya and J. Pandya, 2015).

- CART and C4.5 have similar construction processes. In C4.5, multi-way splits are possible and in CART only binary splits. Furthermore, different pruning methods are used for both (Hu et al., 2014).

- C5 is an extension of C4.5. It is reported that C5 is faster, uses memory more efficiently, and produces smaller decision trees. Additionally, the boosting method is used here (Pang and Gong, 2009; R. Pandya and J. Pandya, 2015).

Like SVM, decision trees are algorithms that can be used for both classification and regression. For regression (if the target variable is continuous), CART uses the *variance reduction* (L. Breiman et al., 1984) as a measure for the split criterion. Regression versions of decision trees are mostly used in connection with EDM to predict grades or grade averages.

The advantage of decision trees over *black-box* classifiers (classification decisions are not easily understood by humans) like SVM is that the decision trees are easier to understand, because the split attributes show why the data is split between branches. In particular, the split attributes used in the top nodes (close to the root) can be used to identify the most important features for predicting the dropout.

The trees can be used to extract production rules (Quinlan, 1987), such as the combination of attribute values that lead to passing or failing in a course (Romero et al., 2008).

2.1.2.2 Random Forest

Since decision trees tend to adapt too much to training data and generalize less well (overfitting), Ho (1995) proposed the random decision forest approach. In this approach, several trees are trained, with each tree trained on a random subset of the available attributes. The output class of the forest can then be chosen as the class predicted by most trees (voting for the majority class). In the following years, various strategies for randomizing the training of the individual trees were proposed to train the trees differently from each other.

Leo Breiman et al. (1996) proposed the ***bootstrap aggregating*** (bagging) approach to create a forest as an improvement over random feature subset selection. In bagging, the training set $L = \{(x_i, y_i)|x_i \in \mathbb{R}^m\}_{i=1,\ldots,n}$ (where x_i are the training points and y_i are the corresponding classes) is used to create a set of m different training sets $\{L_k\}_{k=1,\ldots,m}$. Each L_k is generated from L by sampling with replacement and each L_k contains as many elements as L. If $h(L)$ is a decision tree trained on the training set L, then $\{L_k\}_{k=1,\ldots,m}$ will give us a set of different decision trees $\{h(L_k)\}_{k=1,\ldots,m}$. Domingos (2012) mentions that bagging considerably reduces the variance ("tendency to learn random things irrespective of the real signal") while only slightly increasing the bias ("tendency to consistently learn the same wrong thing").

Leo Breiman (2001) provides an overview of further methods to train different trees and defines the set of different trees as *random forest* (RF). For experiments in Chapter 3, we will use RF, which is based on the bagging approach of CART trees.

2.1.3 Neural Networks

Agrusti et al. (2019) state that the most frequently-used neural network for student dropout prediction is the *multi layer perceptron* (MLP). In this chapter, we briefly introduce the background of MLP, whereby our explanations of neural networks are partly based on Reed and Marks (1999) and Hastie et al. (2009).

Hastie et al. (2009) describe that there was a major hype about neural networks and that many consider them "magical and mysterious", but they are simply nonlinear statistical models. The study of artificial neural networks (ANN) began with a fundamental article by McCulloch and Pitts (1943), in which a simplified model of a (real biological) neuron with multiple inputs and one output was presented.

Slightly later, the perceptron (Rosenblatt, 1958) was introduced, on whose idea the MLPs used today are based. The structure of a single artificial neuron is illustrated in Figure 2.4.

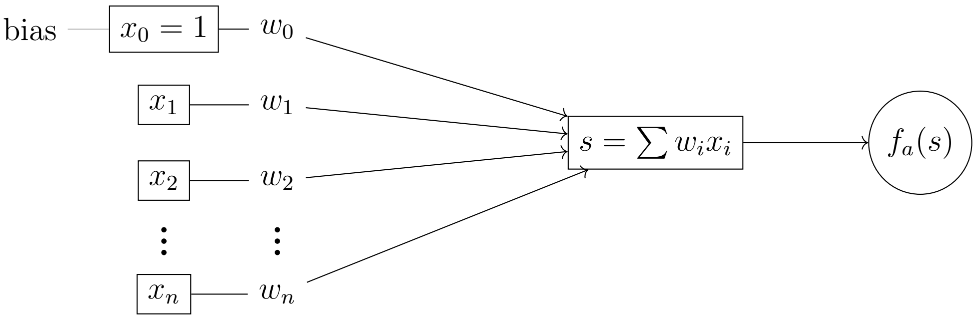


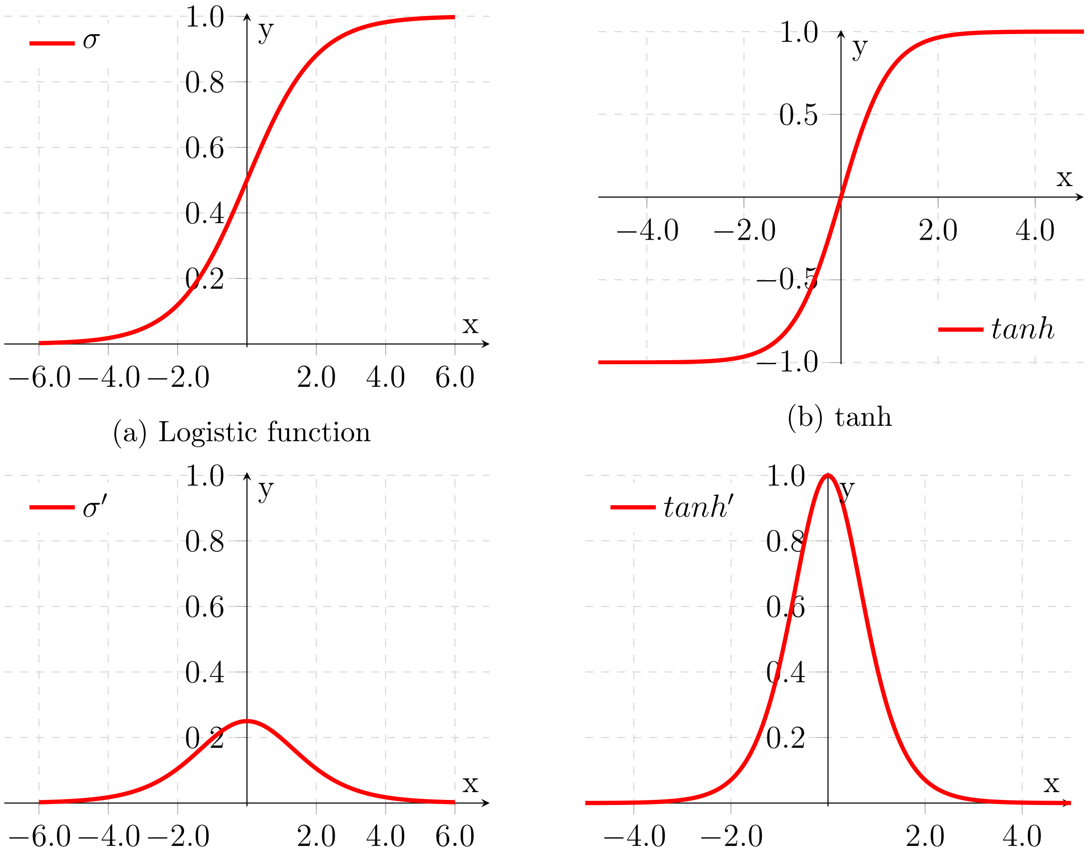
Figure 2.4: Visualization of the inputs and the output of a single neuron

The artificial neurons in a network are also called nodes. A node can have n inputs $\{x_1, ..., x_n\}$, which are summed in a weighted linear combination $\sum w_i x_i$. Often an additional input called a bias with a fixed value of 1 is added. Additionally, there is an activation function f_a, which produces the output value $f_a(\sum w_i x_i)$ when applied to the linear combination. Often sigmoid functions are chosen as non-linear activation functions. Sigmoid functions $s(x) : \mathbb{R} \to \mathbb{R}$ are s-shaped, bounded, and differentiable functions that have a positive derivative everywhere (Han and Moraga, 1995). For $x \to \pm\infty$ sigmoid functions are limited by horizontal asymptotes.

In connection with neural networks, the logistic function $\sigma(x)$ (Equation 2.10) and tanh(x) (Equation 2.11) are often used. The two sigmoid functions are visualized in Figures 2.5a and 2.5b. The derivatives of the two functions (Figures 2.5c and 2.5d), which are important for the training of the network, are defined by $\sigma'(x) = \sigma(x)(1-\sigma(x))$ and $tanh'(x) = 1 - tanh(x)^2$.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.10}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.11}$$



(a) Logistic function

(b) tanh

(c) First derivative of the logistic function

(d) First derivative of tanh

Figure 2.5: Visualization of sigmoid functions (logistic and tanh) and their derivatives

The network, which comprise several neurons, can in principle have any structure, although often networks in layered structure are used (Reed and Marks, 1999). A commonly-used network is the single hidden layer network, which is sometimes called the *vanilla network* (Hastie et al., 2009). It comprises three layers of nodes:

- Input layer: These are neurons that receive the signals or feature values and pass them on to the next layer.
- Output layer: This layer consists of nodes that present the output of the network.
- All inner layers of the network are called *hidden layer*. Since the network is only visible externally through its input and output layers, the term hidden is used.

Figure 2.6 shows a single hidden layer network, which belongs to the class of *feedforward* networks. The output of the nodes in a feedforward network is only passed to the front layers of the network so that the connections of the nodes do not form circles (the class of networks in which circular connections are possible are called recurrent neural networks). The number of nodes in each layer of the single hidden layer network should be at least 1 and may be of any size. The layers of the network are additionally distinguished as active or passive. Accordingly, the input layer is considered passive, because the nodes in this layer do not process anything and only pass on the data to the next layer. If all nodes of two consecutive layers are connected, the network is called *fully connected*. Single hidden layer networks are a special case of the MLP class. The MLP class includes fully-connected feedforward networks with at least three layers and nodes with non-linear activation functions.

Networks that belong to feedforward networks but do not belong to the class of MLP networks are those whose outputs from nodes of a layer L can serve as input for nodes of layer $L + 2$ and thus skip layer $L + 1$. In MLP networks, outputs from nodes of layer L can only be connected to nodes of layer $L + 1$.

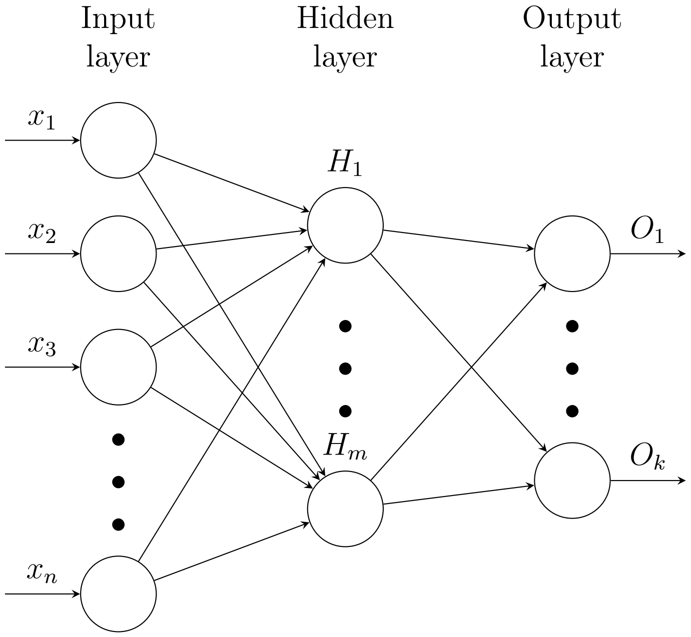


Figure 2.6: Visualization of a single hidden layer neural network

MLP can be used for regression and classification. In a multi-class problem, one would set the number of inputs to the number of features/attributes of the data and the number of output layer neurons to the number of classes.

In a binary classification problem, only one output node can be used and in this output node the logistic function is selected as the activation function. The output value of the sigmoid logistic function is $\sigma(\sum w_i x_i) \in [0, 1]$. Accordingly, the result can be seen as a kind of probability that an object belongs to class 1 (Hastie et al., 2009). By using a threshold value $\theta \in [0, 1]$, the output value of the network produced by the logistic function can be converted into a fixed class assignment. For a new point x^*, the assignment to one of the two classes 0 or 1 is made as in equation 2.12. Usually $\theta = \frac{1}{2}$ is used and moving it up or down can result in higher or lower recall or precision values.

$$f(x^*) = \begin{cases} 1, & \text{if } \sigma(w^T x^*) > \theta \\ 0, & \text{else} \end{cases} \tag{2.12}$$

Cybenko (1989) has shown that every continuous function can be approximated by a suitable single hidden layer network (with some mild assumptions). In literature, this theorem is called the *universal approximation theorem*. However, the theorem does not indicate how many nodes the hidden layer should have and how the appropriate weights can be found.

A solution for training the feedforward networks was presented with the *backpropagation* algorithm (Rumelhart et al., 1986). The idea is to use a cost function based on the difference between the output of the network and the expected value. To minimize the cost function, gradients of the error function are calculated and backpropagation describes how the gradients are calculated and propagated so that the network weights are updated (LeCun et al., 1998).

2.1.4 Discussion

In Figure 2.7, three different binary classification problems are visualized in the column *input data*. The points of the two classes are represented by red and blue.

The first problem (first row) contains two sets of points that are not linearly separable, but only a few points prevent the linear separation. The second problem (second row) is constructed in such a way that one point set is partially enclosed by the other, while the third problem (third row) shows two completely circular point sets where one is completely enclosed by the other and thus no linear separation is possible.

To solve the three classification problems the classifiers SVM (linear kernel), SVM (RBF kernel), decision tree (CART), RF (ensemble of CART trees), and neural network (MLP) are compared. The decision boundaries of the individual classifiers show at which boundaries the class assignment decision is made. The darker the colour, the more reliable the decision. In addition, an accuracy value indicates the percentage of points that were correctly classified.

The SVM with the linear kernel achieves an accuracy value of 88% in the first (simpler) problem, but does not achieve a good classification for the non-linearly separable problems.

The RBF classifier is obviously better adapted to the data and, due to its properties it can separate the data with a non-linear smooth decision boundary, so that an accuracy value of 88% is achieved in the most difficult second problem.

The decision tree separates the space in each node according to a single dimension, so that rectangular decision boundaries are created. The CART tree is also able to adapt relatively well to the data, so that the difficult second problem is solved with an accuracy value of 80%.

The RF, which comprises several trees, has a more finely-structured decision boundary than a single tree and can therefore better approximate the data, whereby it achieves an accuracy value of 85% in the difficult second problem.

The neural network (MLP) with a hidden layer is able to solve the second difficult problem best with an accuracy value of 90% but is worse than the SVM in solving the first problem.

Aside from the SVM (linear kernel), all the classifiers studied were able to solve the third problem perfectly. This investigation shows that there is no classifier that always provides the best solution for each data set. For this reason, often several classifiers are compared on new unknown data sets to select the one that better generalizes the given data (Domingos, 2012).

2.2 Evaluation Measures

In order to measure how well supervised ML methods are able to generalise, evaluation measures are necessary. We will briefly introduce the most commonly-used measures for classification and regression.

2.2.1 Binary Classification Measures

For binary classifiers, the classification result of a data set can be represented with a *confusion matrix* (Table 2.1).

Table 2.1: Confusion matrix

		predicted	
		$\ominus$	$\oplus$
true class	$\ominus$	TN	FP
	$\oplus$	FN	TP

We use it to measure how often elements of a class have been classified correctly or incorrectly. For two classes $\oplus$ and $\ominus$, this results in the following 4 measures:

- True Positive (TP): The number of times class $\oplus$ was actually classified as $\oplus$
- True Negative (TN): The number of times class $\ominus$ was actually classified as $\ominus$
- False Positive (FP): The number of times class $\ominus$ was incorrectly classified as $\oplus$
- False Negative (FN): The number of times class $\oplus$ was incorrectly classified as $\ominus$

From these four measures, the commnonly-used measures *precision*, *recall*, *F*1-*Measure*, *accuracy*, and *false positive rate* (FPR) can be calculated:

- $recall = \frac{TP}{TP+FN}$
- $precision = \frac{TP}{TP+FP}$
- $F1 = 2\frac{Precision \cdot Recall}{Precision+Recall}$
- $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- $FPR = \frac{FP}{FP+TN}$

The value of recall indicates in percent how many elements belonging to class ⊕ were classified as ⊕. The value of precision gives the percentage of how many elements classified as ⊕ are really ⊕. F1 indicates the harmonic mean of precision and recall. The accuracy indicates the ratio of correctly classified ⊕ elements to all elements.

Recall, precision and F1 were defined so that their values are valid for the ⊕ class. Of course, they can also be simply calculated for the ⊖ class. In classification problems, therefore recall, precision and F1 are often specified for each class.

If a binary classifier f outputs a probability $f(x) \in [0,1]$ for a class instead of a class assignment, a threshold θ can be used such that the final class assignment is specified by:

$$F(\theta, x, f) = \begin{cases} 1, & \text{if } f(x) \geq \theta \\ 0, & \text{else} \end{cases}$$

Different visualizations like the *receiver operator characteristic* (ROC) curve or the *precision-recall* curve can be used to visualize the evaluation for different thresholds (Davis and Goadrich, 2006).

The ROC curve is a plot of recall against FPR, whereby each point of the curve represents the recall and FPR result for a given threshold θ. The ***a**rea **u**nder the **c**urve* (AUC) score is often used to compare the ROC curves of different models. This score summarizes the ROC curve and is sometimes called ROCAUC.

Davis and Goadrich (2006) state that in case of high class imbalance (one class has much more data points than the other class) the precision-recall curve, which is a plot of recall against precision, is more suitable. Each point of the precision-recall curve represents the recall and precision result for a given threshold θ. Similar to the ROCAUC score, the precision-recall curve can be summarized by the *average precision* (AP) value.

2.2.2 Regression Measure

If $f : \mathbb{R}^n \to \mathbb{R}$ is a regressor and $D = \{(x_i, y_i)_{i=1,\ldots,m} \in \mathbb{R}^n \times \mathbb{R}\}$ is a set to evaluate f, then the *root mean square error* (RMSE) (Géron, 2018) is defined by

$$RMSE(f, D) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2}$$

.

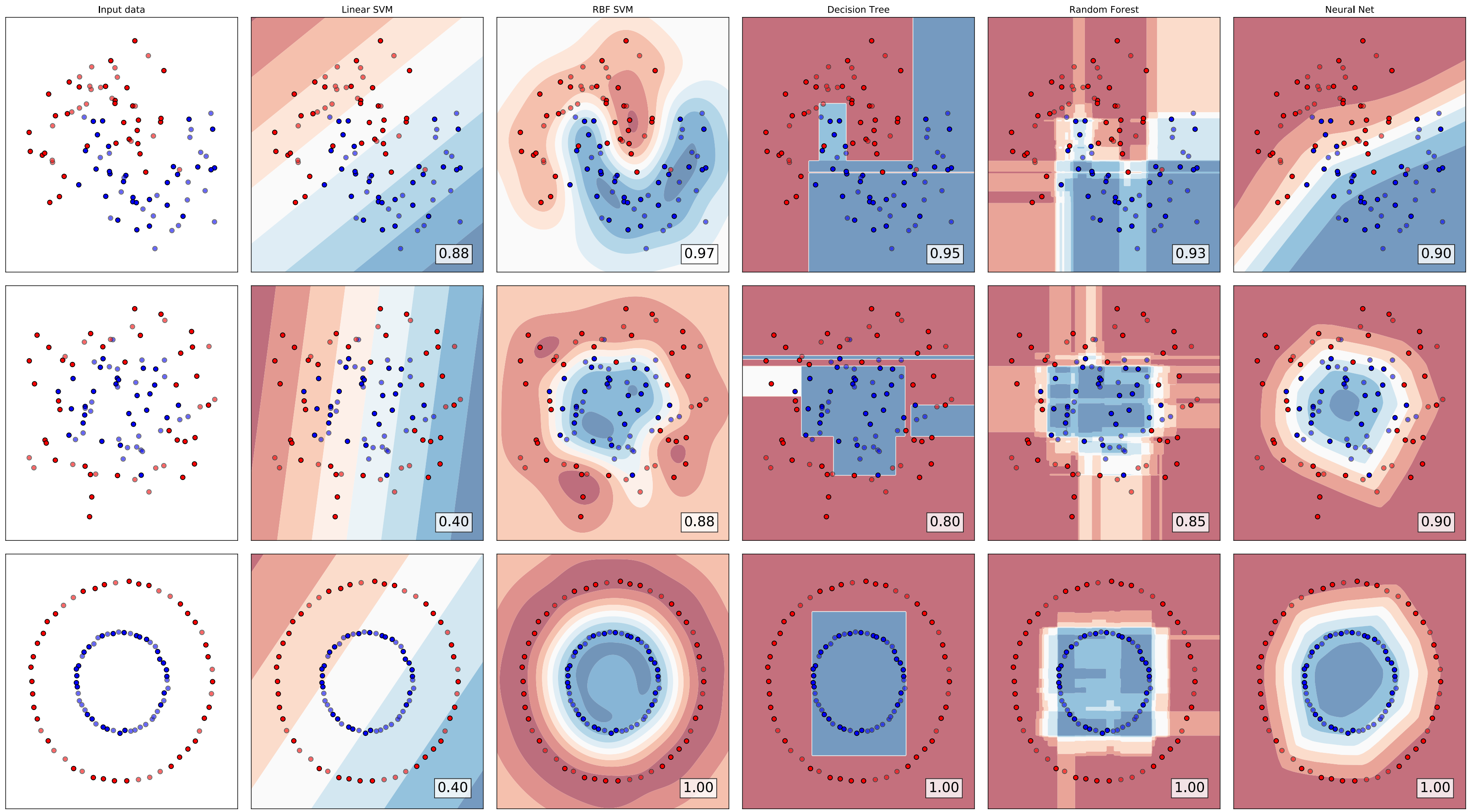


Figure 2.7: Comparing classifier boundaries[a]

[a] The visualization is motivated by `http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`

3

Student Performance and Dropout Prediction

"It is very difficult to predict — especially the future."
— Niels Bohr (Mencher, 1971)

One of the most active research topics in EDM and LA is the prediction of student dropout and student performance (Aldowah et al., 2019). In order to help students before they drop out of a course or the whole study, they need to be identified early (Romero and Ventura, 2019). Therefore, we use data mining and machine learning algorithms for early prediction.

To apply data mining and machine learning to student data, the raw data which could be in any format (e.g., server logs) and, therefore, not easy to analyze must be transformed into a suitable format. Most algorithms require a n dimensional numerical representation $x = (x_1, ..., x_i, ..., x_n) \in \mathbb{R}^n$.

In this vector representation, the students are modeled by their different properties where values x_i describe different properties of the students. The properties used to create the vectors are called attributes or features (or covariates) (Murphy, 2012).

To create the vectors, in a first step, useful features must be determined from the raw data. The art of creating features from raw (and partially unstructured) data is called *feature engineering*. On student data, for example, simple features such as age or previous grades can be used. In a further step called *feature extraction*, new features can be created from the initial features (Géron, 2018). For example, if a student's initial features include previous grades, average grades can be calculated as new features. These features can either be determined by an expert or extracted from the data by automatic procedures.

Before ML algorithms can work with the feature vectors, different transformations and preparation steps have to be done with the data, which are discussed in Chapter 3.1.

In Chapter 3.2, we look at different feature types used in the EDM area. Depending on the given tasks, these features can vary greatly, e.g., predicting grades in a course may require different features than predicting dropout. In the following chapters on dropout prediction (Chapter 3.4) and student performance prediction (Chapter 3.5), we will discuss the respective feature sets.

If existing features are not sufficient, new features can be created by collecting new data. Today, there are more and more digital learning environments that offer the ability to track student data. In Askinadze and Conrad (2017), we investigated how information from such distributed learning environments can be tracked and integrated, which we briefly discuss in Chapter 3.3.

3.1 Preprocessing

For the creation of the features and their transformation into the final representation of the students as numerical vectors, different *preprocessing* steps are necessary.

Especially in the EDM area, the data can be very heterogeneous. For example, it could be time-series data of clicks in a digital learning environment or demographic data like the age of the student. A property like age is a numerical feature because this feature is described with numerical values. Features that have a finite set of non-numerical values are called categorical features, i.e., the categorical feature *gender* has the two possible values *male* and *female*. In Chapter 3.1.1, we discuss the transformation of categorical features into numerical features.

Features can have very different values (e.g., the number indicating the annual salary is much higher than the number indicating a student's age). Since different ranges within the features can have negative effects on ML algorithms, we outline the two most commonly used scaling approaches in Chapter 3.1.2.

Another important step is *data cleaning* since data may contain implausible outliers or missing values. Since most ML procedures cannot work with missing values in feature vectors, imputation strategies are necessary, which are discussed in Chapter 3.1.3.

Finally, it is very important to select from the available features those that are relevant to the given task. In Chapter 3.1.4, frequently used feature selection procedures are presented.

3.1.1 Transforming Categorical Features

For use in most ML algorithms, the categorical features have to be transformed into numerical ones. To indicate the possible transformations, we distinguish two types of categorical features:

- **nominal**: Nominal features have a finite number of possible values that have no numerical representation and are not subject to any order, i.e., gender with the two options "male" or "female" or the country of origin. If the set of values of a feature consists of only two options, then the value 0 or 1 can be assigned to both values, i.e., male = 0 and female = 1. If there are more than 2 possible values, one-hot-encoding (VanderPlas, 2016) can be used.

- **ordinal**: Ordinal features are nominal features that can be arranged in natural order, e.g., the feature "English knowledge" can have the three values low, middle, high. A numerical value can be assigned to each of these characteristics: low=0, middle=1, high=2.

3.1.2 Scaling Features

If, for example, we only know the age of a student “20”, the gender “male” (with the coding female=0, male=1), and English knowledge “high” (with the coding low=0, middle=1, high=2), the numerical representation of the student could be given by $(20, 1, 2)$. Depending on the algorithm, it can lead to problems if the features are not scaled (especially with distance-based classifiers like the RBF SVM). The feature with the largest range (e.g., the age in the example above) then dominates the other features, although it does not have to be the most important one. This is not true for all algorithms. For example, decision trees can handle the data well without scaling, if the split is performed by single features. The following two approaches are mostly used in preprocessing depending on the type of data:

- **Standardization**: Removing the mean and dividing by the standard deviation of the training samples. The process is performed per feature individually so that the features are centered around zero and have a standard deviation of one.
- **Rescaling**: Features are rescaled to the interval [0,1] (min-max scaling) or [-1,1].

3.1.3 Missing Value Imputation

Most classifiers or regression algorithms require that all features are available in a feature vector. However, some features may not be present in the data for various reasons. There are several ways to deal with missing features:

(a) Removing features (columns) whose values do not always appear in the student data. For example, if the values of a feature f_1 are missing in the data of only 1% of the students, the feature f_1 would not be usable for all other students either. Since this can lead to removing relevant features, this approach is not commonly used.

(b) Remove students (rows) who have missing features. If students are removed from the training data sets, less data will be available for training. Since student data sets often are already small, this is not a reasonable approach.

(c) The most reasonable approach is to estimate the missing values. There are several ways to do this:

  - a fixed value (for example, the average of a characteristic) can be used for the missing values
  - Adaptive approaches, such as k-NN imputation, can be used to estimate missing characteristics by averaging the characteristic values of k similar students (where the values of the missing characteristic are known).

  In Askinadze and Conrad (2018b), we investigated for a public available data set[1] (Cortez and Silva, 2008) how missing feature values affect the prediction of student performance. For 25% of missing data, using k-NN to estimate the missing values yields evaluation results that are not much worse than using the full data set. As the number of missing values increases, the prediction results become significantly worse.

[1] https://archive.ics.uci.edu/ml/datasets/student+performance

3.1.4 Feature Selection

In the literature, classifiers are often compared on all data, so that no examination of individual feature sets is made (Aulck et al., 2019). The art of selecting the appropriate features for a given task is called *feature selection* and is a very active research topic in ML (Murphy, 2012).

We first consider the reasons why it makes sense to select a subset of the existing features. Guyon and Elisseeff (2003) give the following motivation for feature selection in their work:

- "Facilitating data visualization and understanding"
- "Reducing training utilization times"
- "Reducing the measurement and storage requirements"
- "Defying the curse of dimensionality to improve prediction performance"

In the research field of EDM, we can add *preserving data privacy* as another important reason. Especially for feature selection, the principle of *data minimization* is important. According to Art. 5 1(c) of the GDPR (European Union, 2016), the minimization of the storage of personal data to the necessary extent is necessary. If we consider a subset of features to be sufficient to make as good predictions as if all features were used, we could stop storing data for the features that are not required and thus meet the requirements of the GDPR.

To select subsets of features, one should first define what good or bad features are. Definitions of relevance and irrelevance of features are given in Kohavi and John (1997) and Blum and Langley (1997). Roughly speaking, features are *strongly relevant* if they contribute information to a given ML task that other features do not. Features are *weekly relevant* if they are not strongly relevant and carry information that is also present in other features. Features are irrelevant if they are neither strongly nor weakly relevant.

Guyon and Elisseeff (2003) provide a good overview of different methods and techniques for feature selection. The methods can be divided into three classes *filters*, *wrappers*, and *embedded methods* which are briefly described in the following Chapters.

3.1.4.1 Filter Methods

Filter methods select features in a preprocessing step, independent of a classifier (Kohavi and John, 1997). The selection is done by ranking the features based on a measure of quality. Feature filtering is also called screening or ranking (Murphy, 2012).

3.1.4.1.1 Correlation

The *Pearson correlation coefficient* ρ for two feature vectors $x, y \in \mathbb{R}^n$ is a measure of linear dependency and is defined by:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

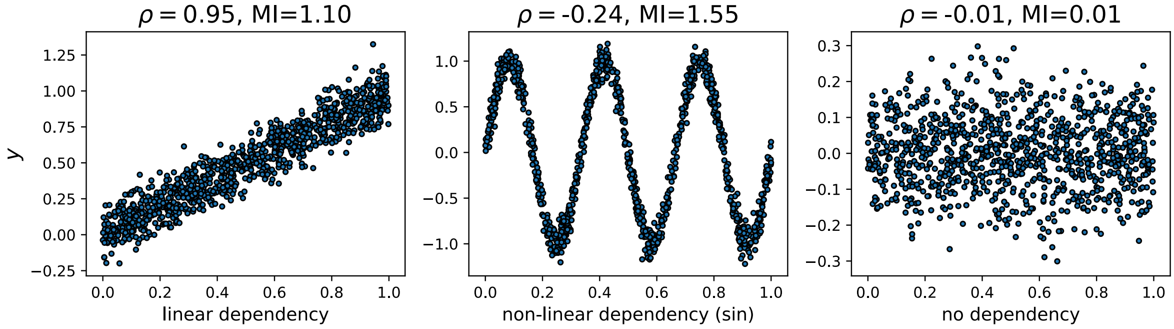


Figure 3.1: Comparing correlation coefficient and MI score[2]

It does not consider non-linear dependencies (Guyon and Elisseeff, 2003; Smith, 2015). Assuming a linear dependency between the features and the output variable, the correlation of the features and the output variable can be used to measure for which features a higher linear dependency exists and to rank the features depending on the correlation coefficient.

3.1.4.1.2 Mutual Information

One of the disadvantages of the correlation coefficient is the fact that only linear dependencies can be found with it. An approach that can also find non-linear dependencies is *mutual information* (MI) (Smith, 2015; Dionisio et al., 2004). Guyon and Elisseeff (2003) and Murphy (2012) show how the relevance of a discrete or nominal feature i to class label Y based on mutual information is calculated. MI can also be used for continuous features and outcomes. Figure[2] 3.1 shows for the three cases (a) linear dependency (b) non-linear dependency and (c) no dependency between input and output, which scores are obtained by Pearson correlation (ρ) and MI. For the Pearson correlation coefficient $\rho \in [-1, 1]$, values near 1 and -1 means a strong linear dependency and values near zero means no linear dependency. Mutual information takes only non-negative values and the higher the dependency, the higher the MI score. In case (a), the correlation coefficient with the value of 0.95 shows a strong linear dependency. In case (b), the correlation coefficient shows only a weak dependency (-0.24), while the MI score finds the non-linear dependency and rates it even higher than in case (a). In the case of no dependency, both scores are close to 0.

3.1.4.1.3 Two-sample t-test

In a two-class problem, such as the separation between students with good and bad grades, the *two-sample t-test for equal means* can be used to find suitable features (Chandra and Gupta, 2011). Other variants like the Welch test can also be used depending on the nature of the data (Ruxton, 2006). The t-test is a hypothesis test to examine if the assumption that two population means are equal can be rejected. The hypothesis that the means of two populations are equal can be rejected if the calculated p-value is less than a chosen significance level threshold (common values are 1%, 5%,

[2] Figure motivated and adapted from `scikit-learn.org/stable/auto_examples/feature_selection/plot_f_test_vs_mi.html`

or 10%). If the hypothesis can be rejected, then the feature can be included in the resulting feature set.

3.1.4.1.4 ANOVA

In a multi-class problem, a popular method in psychological and educational research (Blanca et al., 2017), *the one-way **analysis of variance*** (ANOVA) can be used for feature selection. ANOVA is a statistical technique based on the F-test to compare the means μ_i of $n > 2$ populations. ANOVA examines the null hypothesis $H_0 : \mu_1 = ... = \mu_n$ against the alternative hypothesis H_1: *at least one pair of the means is unequal.* For $n = 2$ ANOVA is equivalent to the t-Test. Kim (2014) pointed out that it is incorrect to perform multiple t-Tests on multiple pairs of averages, in the case of more than two populations and that ANOVA is the appropriate option. Using the ANOVA technique, an F-value is calculated per feature. Then, the features are sorted by their F-value and, finally, an appropriated subset of features with the best F-values is selected.

The F-test has some assumptions about the underlying data. Blanca et al. (2017) have shown that ANOVA is robust to various violations of the assumptions (deviation from a normal distribution, sample size, and unequal distribution in the groups) regarding the Type I error.

3.1.4.2 Wrapper Methods

Another class of feature selection methods are the *wrappers.* The wrapper methods compare multiple feature sets based on their usefulness for a classifier. Since the number of possible subsets grows exponentially with the number of features, finding the best subset is an NP-hard problem (Guyon and Elisseeff, 2003). Therefore, greedy approaches such as *sequential forward selection* (SFS) can be used where in an iterative process, features that improve the classifier performance are added to the final feature set (which is empty at the beginning). In Askinadze et al. (2018), we used a combination of filter and wrapper methods since we first created several feature sets with filter methods and finally tested the feature sets on several classifiers to find a good combination of feature set and classifier.

3.1.4.3 Embedded Methods

In embedded methods, the features are selected during the training of a classifier. For example, decision trees belong to embedded methods (Guyon and Elisseeff, 2003).

3.2 Student Feature Categories

Recent studies conducted experiments with a large number of different features for modeling students. In a recent literature review (Alban and Mauricio, 2019a), a broad list of different features was given with a reference to respective articles. Saa et al. (2019) have examined 34 articles and identified 215 different features. Alban and Mauricio (2019a) categorized student features in 5 dimensions: personal, institutional, economic, academic, and social. Saa et al. (2019) categorized the features into 9 dimensions: students' e-learning activities, students' previous grades and class performances, students' environment, students' demographics, instructor attributes, course attributes, students social information, course evaluations, and students experience information. In Amrieh et al. (2016), the features were divided into 4 categories: demographical features, academic background features, parents participation features on the learning process, and behavioral features. The individual categories overlap or, in our opinion, are subcategories of each other, so that we have structured the different categories in a hierarchical view (Figure 3.2) with two main categories *academic features* (all features resulting from the academic context) and *personal features* (all others).

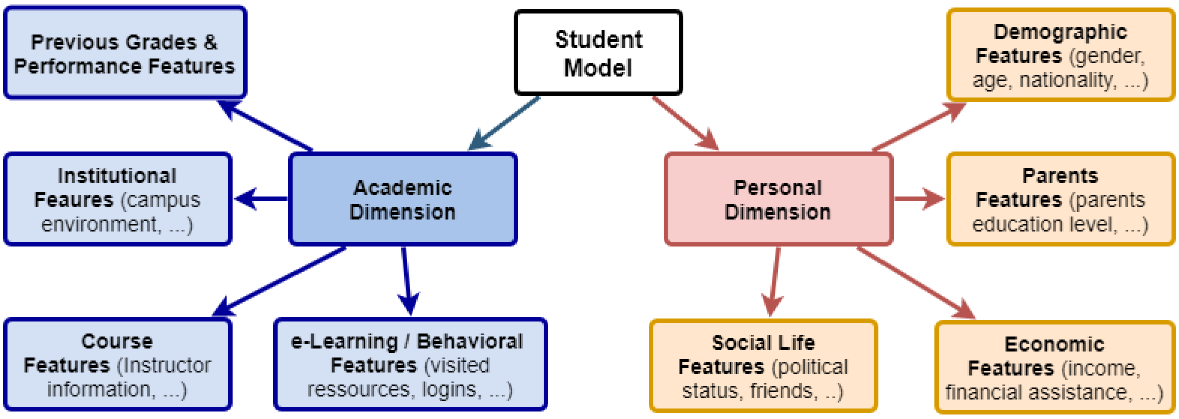

Student Model
Academic Dimension
Personal Dimension
Previous Grades & Performance Features
Institutional Feaures (campus environment, ...)
Course Features (Instructor information, ...)
e-Learning / Behavioral Features (visited ressources, logins, ...)
Demographic Features (gender, age, nationality, ...)
Parents Features (parents education level, ...)
Social Life Features (political status, friends, ..)
Economic Features (income, financial assistance, ...)

Figure 3.2: Hierarchical visualization of feature categories used for student modeling based on Alban and Mauricio (2019b), Saa et al. (2019), and Amrieh et al. (2016)

Saa et al. (2019) examined which feature category is mostly used to predict student performance in higher education. They found out that the following three feature categories occur most frequently in descending order: previous grades and performance, e-learning activities, and demographics.

The reason why these three categories are used is most likely because they are most often available to the respective research teams. Demographic information such as age and gender is usually always available to educational institutions, as well as information about exams already passed within the institution. In fact, as we will see later on, information on past academic performance is one of the most important predictors.

To obtain information about a students' social life, surveys would have to be conducted or, for example, the social relationships of students would have to be analyzed by examining the communication between students (Bayer et al., 2012).

Various studies (Tross et al., 2000; Komarraju and Karau, 2005; Conard, 2006; Komarraju et al., 2009; Komarraju et al., 2011) have examined the influence of the "big five" personality traits (*openness, conscientiousness, extraversion, agreeableness, and neuroticism*) on academic performance and motivation and have shown that these traits can be used as predictors. Universities do not normally have such data about their students. In a feature extraction process, such features could be predicted from other previously known features. In the case of students of a programming course, we have investigated in Liebeck et al. (2016) how the big five personality traits can be predicted from program code. With the proposed features, we have participated in a challenge (Rangel et al., 2016) and were better than the median for four of the five personality traits in terms of RMSE and were able to achieve the lowest RMSE value for conscientiousness with our approach.

E-Learning systems are another way to collect additional data. If the institutions use e-learning systems, the clickstream of these systems can be tracked. In the following chapter, we examine an architecture for tracking such student interactions.

3.3 Student Data Tracking

In the previous chapter, we have seen different categories of features that have been used to predict student performance. The sources of data for these features can be very heterogeneous, i.e., the demographic data and exam data may be stored in a central administration database. In Askinadze and Conrad (2018a), we examine how the study history data of multiple universities can be stored in a shared system. Individual study courses could use different e-learning services from different third-party providers so that the usage data of the students are distributed in different databases and systems.

In Askinadze and Conrad (2017), we investigated how this can be achieved. The idea is based on the use of the Experience API (xAPI[3]) specification. This specification describes a data format for storing student interactions in digital learning environments and transferring them to other systems using a REST-API. Figure 3.3 shows the xAPI architecture, in which several learning tools send student interactions in xAPI format to the xAPI interface, where they are stored in a database (*learning record store*). Finally, the stored xAPI statements can be aggregated and displayed in a dashboard, or new knowledge can be extracted from the data using EDM methods to make predictions about students, which can then also be displayed in a dashboard.

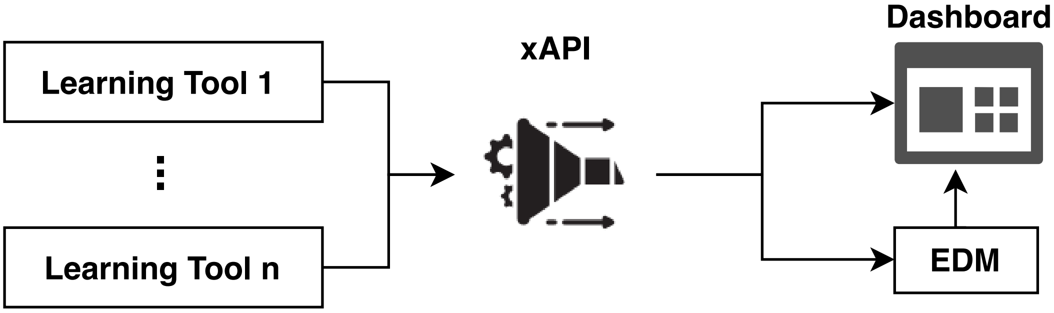


Figure 3.3: xAPI Tracking Architecture

The data format itself describes the answer to the question "who did what?" so that a JSON object with at least three properties (subject, verb, and object) is stored. For example, if a student has browsed back in an eBook, the subject has the information about the student (e.g., an id to identify the student). The verb has the information which action the student has performed (in this case, "clicked back") and the object has the information about the eBook and on which page in the eBook the interaction took place. xAPI is already used in many LMS and learning software like h5p[4].

The example with the eBook was chosen here because in Chapter 3.5.2 we show in a case study how student performance can be predicted based on xAPI usage data in an eBook. In Chapter 3.5.1 xAPI data from a learning management system is used to predict final grades in a course.

[3] `https://github.com/adlnet/xAPI-Spec`

[4] `https://h5p.org/`

3.4 Dropout Prediction in Higher Education

The dropout of studies has been researched for a long time. Tinto (1982) investigated the *completition rate and dropout rate* of American BA students from 1880 to 1980 and found out that the dropout rate remained constant at about 45% in the 100 years studied (except during and shortly after World War II). In a more recent study (Schneider, 2010) on dropout rates in the USA, it is reported that 30% of first-year students do not come back after one year and that this dropout group causes an estimated annual cost of $1.5 billion. Besides the financial damage to the public, dropout also has negative effects on the students themselves, as they lose time (if they have not learned anything useful for themselves in that time) and could get negative feelings like self-doubt because of the dropout (Larsen et al., 2012).

In the acatech study (Klöpping et al., 2017), the number of dropouts in engineering courses in Germany was examined based on data from 12 universities. The authors have found out that after six semesters, there are about 6% of students changing to another subject, 10% of students changing to a different institution of higher education, and 21% dropouts. The numbers fluctuate only slightly between the cohorts studied.

Berens et al. (2019) mention that there are many programs in Germany to reduce the drop-out rate at universities, but that students would have to apply to the respective programs themselves. The authors suggest that administrative data from universities should be used to find students in danger so that these students can be more effectively supported. Hartl (2019) discussed how Data Mining can support university management.

The recent review papers of Agrusti et al. (2019) and Alban and Mauricio (2019b) give a good overview of often used classifiers, preprocessing techniques, and features in this particular research area.

In the remainder of this chapter, we present an approach to the prediction of student dropout, which we evaluate on student data from a German university. The proposed approach requires a relatively small number of students to train the model and includes a minimal feature set (exam results only) so that any university can implement this approach and make predictions about their students at the end of a semester.

3.4.1 Dropout Definition

In literature, there are several possible definitions for dropout. Spady (1970) offers two definitions for dropout: The first is "dropout includes anyone leaving a college at which he registered" (without a degree). The second is "dropout refers only to those who never receive a degree from any college". Klöpping et al. (2017) state that often, no distinction can be made between university changers and university dropouts and uses a definition of university dropouts in which dropouts refer to students who leave the university without a degree and do not actively state a university change as a reason.

In general, the definition depends on the target group that is interested in the evaluation. For the administration of a study program, dropouts are persons who leave the course of study without a degree. The university administration might be more interested in a definition in which the student leaves the university without a degree. For the further usage of the term *dropouts*, we use a definition similar to the first version of (Spady, 1970), which we define as follows: *Dropouts are students who leave the university in which they started their studies without a degree.*

3.4.2 Features for Dropout Prediction in Higher Education

In the literature, many different reasons for dropping out are reported. In the first semesters (in which most drop-outs occur), the reasons are rather a lack of motivation to study or performance problems. In the later semesters, on the other hand, the reasons are often illness, exam failure, and financial or family problems (Fleischer et al., 2019; Heublein et al., 2007). Fleischer et al. (2019) also list other predictors (cognitive-motivational) from the literature that are suitable for predicting study success. These include, for example, "ability of self-assessment" and "conscientiousness".

In general, the reasons for dropping out are complex and there are usually several factors in common that lead to dropping out (Larsen et al., 2012). It is problematic that the data on motivation or health, for example, are usually not available to universities (Berens et al., 2019) or only have to be collected with much effort (e.g., through surveys).

In order to make dropout prediction, the data that is actually available in the university databases must be used in practice. To do this, we can look at the data that universities store because they are required to do so by law. These data differ from country to country. Berens et al. (2019) say, for example, that in Germany, there is the *Higher Education Statistics Act* (HStatG), which obliges universities to store certain demographic and academic achievement data (§3). Hartl (2019) mentions that all German universities have student and applicant data at their disposal. Since the feature sets of student dropout studies often consist of a combination of different demographical and academic features, it is not easy to compare the results of the different studies.

The timing of the prediction determines the available features. Similar to Hartl (2019), we roughly distinguish the following three points in time:

1. **Before the start of studies (or shortly after the start of studies)**

   For the prediction, data on admission to higher education or demographic data could be used. Larsen et al. (2012) mentions that prior academical achievement is a good predictor. Prior academical achievement data is, for example, the overall grade of the Abitur (German term for high school graduation grade), which is the predictor with the highest predictive power for all subjects (Trapmann et al., 2007; Fleischer et al., 2019). Especially in natural science subjects, there is a comparatively high drop-out rate, and for these subjects, prior mathematical knowledge is a good predictor for academic success (Müller et al., 2018; Fleischer et al., 2019). Sclater et al. (2016) have proposed that additional data should be obtained through surveys on emotional and financial status. Ortiz-Lozano et al. (2018) used the admission test grade for prediction and could reach an accuracy value of about 61%. Berens et al. (2019) used the entire demographic data required by the HStatG (and some other data calculated by feature extraction) to achieve an accuracy value of about 67%.

2. **During the semester**

   The prediction becomes possible during the semester if results of different homework or usage data from the digital learning environments arrive gradually. Which data is collected depends on whether the universities offer e-learning with data collection and whether this data is stored at a central location so that a prediction can be made on the data from different sources. In Ortiz-Lozano et al. (2018), the midterm tests data in the first semester was used to predict dropout, which was possible with an accuracy value of about 71%. However, most of the data collected during a semester is used to predict the grade or performance of a course. We discuss this topic in Chapter 3.5) in more detail.

3. **At the end of a semester (or at the end of the exam phase)**

   Prediction can be made here when all exam results are available. This data is available to all universities and is usually stored centrally in an administrative database. Since many students drop out of their studies after the first semester, it is too late for these students to offer help after the prediction. These students should, therefore, be identified by making predictions based on data given in the other two points in time mentioned above.

In the following, we consider the third prediction time type, i.e., when the exam results are available at the end of a semester.

Of particular interest are studies that carry out the evaluation on different feature sets. Dekker et al. (2009) reported that the most important features for student dropout prediction are collected at universities themselves (achievement and performance data). Recently published studies also provide similar results. Aulck et al. (2019) have investigated various data subsets (demographic data, department-level data, First-Year Summary Data, Grouped Course Data, Major Data, and Pre-Entry Data) and found that *student progress based features* are better suited than the pre-entry and demographic information of students. The authors note that the first-year summary data has delivered almost as good results as the use of the entire data set.

Berens et al. (2019) have used the data required by HStatG to predict dropout. Once using the whole data set (including demographic data) and once only the data based on academical achievements and also reported that academical achievements produce almost as good results as the whole data set. Hartl (2019) also reports that pure demographic data cannot be used to make good predictions, but as soon as grades are used, the results can be significantly improved. This is also confirmed by the results in Manrique et al. (2019), which have shown that features based on a small number of grades of important courses are sufficient. The authors have shown that the way the feature vectors are created has a large influence on the classification results. They distinguished the feature vector creation into *Global Feature-Based Representation* (GFB) and *Local Feature-Based Representation* (LFB). Features of a GFB representation summarize a student's data, such as the number of exams passed or grade averages, without describing the data of individual courses. With GFB, the students are described by a feature vector, which looks the same for students of different courses. The LFB representation, on the other hand, uses features that reflect course-specific information. Manrique et al. (2019) have used the grades of individual courses in their LFB approach. The LFB based approach has produced better results on their data set.

In our study (Askinadze and Conrad, 2019), we have proposed to simplify the features even further. Our LFB based approach only uses binary features with the information whether or not the exam for a course has been passed, and also uses only a small set of exams to create the feature vector. In Chapter 3.4.3, we explain the proposed representation and show its use with neural-network-based classification in Chapter 3.4.4.1 and with SVM based classifiers in Chapter 3.4.4.2. In Chapter 3.4.5, we present our results of student dropout prediction, compare them with other studies, and finally discuss the limits of our prediction approach.

3.4.3 Proposed Student Representation

In Askinadze and Conrad (2019), we have introduced a feature representation based only on the information whether students have passed exams or not. In the following, we briefly describe the idea behind this approach. Let $C = \{c_1, ..., c_n\}_{n \geq 1}$ be the set containing the exams that we use to model the student representation. Then the vector for student s representing available data at the end of semester $t \leq k$ is defined by:

$$\psi^C_{t \leq k}(s) = \Big[\underbrace{\quad 0 \ or \ 1 \quad}_{c_1 \ passed \ \textbf{until} \ \text{sem.} \ k} \quad ... \quad \underbrace{\quad 0 \ or \ 1 \quad}_{c_n \ passed \ \textbf{until} \ \text{sem.} \ k} \Big]$$

We use an example to show how to use this representation. Let C be a set of the three exams with c_1 = Calculus, c_2 = Linear Algebra, and c_3 = Programming. Let s_1 be a student who passed Programming in the first semester, Linear Algebra in the second semester, and Calculus in the third semester. Let s_2 be a student who passed Calculus in the first semester, Programming in the second, and Linear Algebra in the third semester. Then the corresponding vector representations are as follows:

$$\begin{aligned} \psi^C_{t \leq 1}(s_1) &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \\ \psi^C_{t \leq 2}(s_1) &= \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \\ \psi^C_{t \leq 3}(s_1) &= \Big[\underbrace{1 \quad 1 \quad 1}_{c_1 \ \ c_2 \ \ c_3} \Big] \end{aligned} \qquad \begin{aligned} \psi^C_{t \leq 1}(s_2) &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ \psi^C_{t \leq 2}(s_2) &= \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \\ \psi^C_{t \leq 3}(s_2) &= \Big[\underbrace{1 \quad 1 \quad 1}_{c_1 \ \ c_2 \ \ c_3} \Big] \end{aligned}$$

This representation is LFB because it uses course-specific data (exams passed). It does not depend on grades (which could be missing if exams are not graded), so there are no issues with missing values. Due to the simplicity, the reasons for classification decisions can be better understood, e.g., rules can be derived by rule induction approaches (Hartl, 2019). Rule extraction has already been applied for performance prediction in a course (Al-Radaideh et al., 2006; Cortez and Silva, 2008; Hu et al., 2014).

In a case study on the dropout behavior rule extraction of computer science first-semester students, we train a decision tree (CART with entropy-based split criterion) based on our student representation. As features, we use the information, whether the students have passed the first-semester exams Calculus, Linear Algebra, and Programming at the end of the first semester. The corresponding tree is visualized in Figure 3.5. The *value* attribute in the nodes shows the distribution of the two classes graduate and dropout after the split. The selection of a feature for a split is performed based on entropy. The feature that separates the students notably well on the two classes graduate and dropout is the information whether the students have passed the Calculus exam. By looking at the root node, we can see which exam is the best predictor for success. Since the features are binary, the decision for a split is made using the threshold 0.5, resulting in notations such as $Calculus \leq 0.5$ in the nodes. The color represents how strongly the majority class is represented in a node. From the 8 leaves of the tree, we can see that some paths of the tree lead to relatively safe decisions, like if all 3 exams are passed or if none of the exams are passed. Some decisions of the tree are not a good basis for decision making, as in the case of $Calculus = 0$ & $Programming = 1$ & $LinearAlgebra = 0$. If the students have passed Programming but not a single math exam after the first semester, then 50% of the

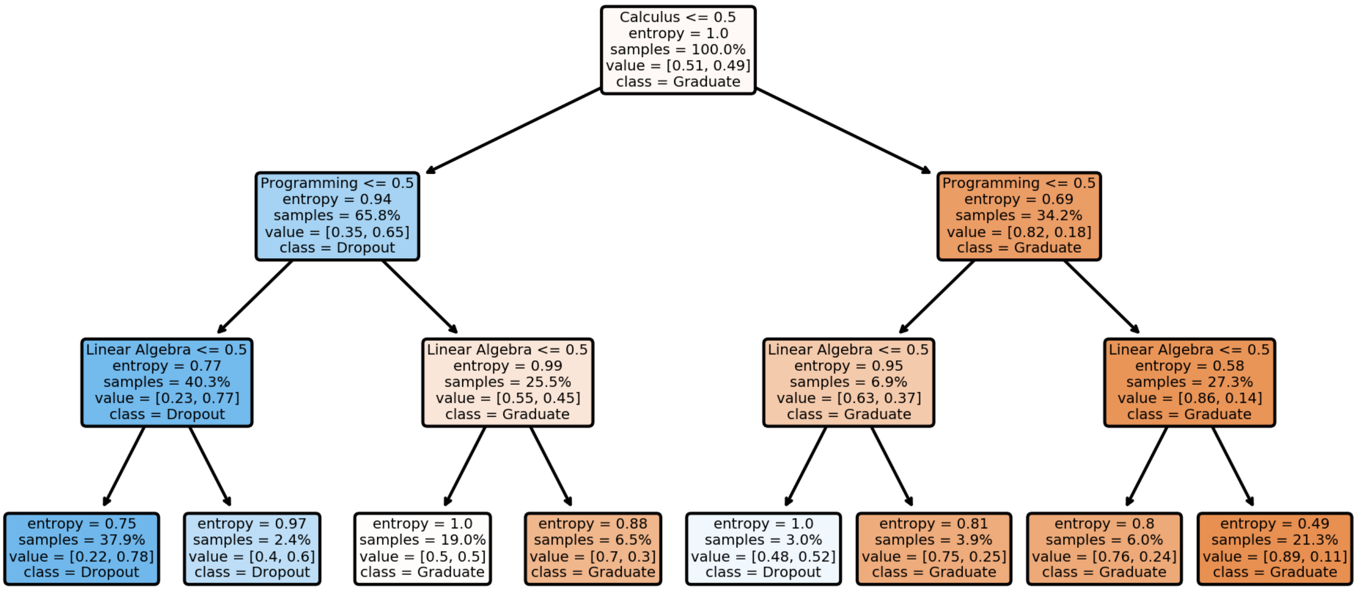


Figure 3.5: Dropout prediction after the second semester using a decision tree

students in this cohort will drop out in the future. On the other hand, passing the two mathematics exams (especially Calculus) in the first semester is a good predictor for a successful study. In total, the following 5 rules can be extracted from the 8 leaves:

1. $if(Calculus = 1 \;\&\; Programming = 1) \rightarrow Graduate$
2. $if(Calculus = 1 \;\&\; Programming = 0 \;\&\; LinearAlgebra = 1) \rightarrow Graduate$
3. $if(Calculus = 1 \;\&\; Programming = 0 \;\&\; LinearAlgebra = 0) \;\; \rightarrow Dropout$
4. $if(Calculus = 0 \;\&\; Programming = 0) \rightarrow Dropout$
5. $if(Calculus = 0 \;\&\; Programming = 1) \rightarrow Dropout$

In contrast to the proposed LFB representation, a corresponding GFB representation that simply indicates the number of exams passed $GFB^C_{t \leq k}(s) = ||\psi^C_{t \leq k}(s)||_1$ would be equal for both students $GFB^C_{t \leq 2}(s_1) = 2 = GFB^C_{t \leq 2}(s_2)$ while $\psi^C_{t \leq 2}(s_1) \neq \psi^C_{t \leq 2}(s_2)$.

However, in the case of the third semester, both representations are also equal in terms of distance, since $\psi^C_{t \leq 3}(s_1) = \psi^C_{t \leq 3}(s_2)$ and $GFB^C_{t \leq 3}(s_1) = 3 = GFB^C_{t \leq 3}(s_2)$. Therefore, we have proposed to represent a student not only by the last known state of the exam results but by the whole time series instead which we define by $T^C_k(s) = \left[\psi^C_{t \leq 1}(s), \ldots, \psi^C_{t \leq k}(s)\right]$. For the example of the two students s_1 and s_2 we get two unequal time series $T^C_3(s_1) \neq T^C_3(s_2)$ which are visualized as follows:

$$\underbrace{\left[\begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix}\right]}_{=T^C_3(s_1)} \qquad \underbrace{\left[\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix}\right]}_{=T^C_3(s_2)}$$

3.4.4 Applying Classifiers to our Representation

For the evaluation of the dropout prediction, the classifiers RF, MLP, and SVM with different kernels will be compared. While applying RF is straightforward, we show in this Chapter how MLP and SVM are applied using the proposed student representations.

3.4.4.1 Neural Network based Student Dropout Prediction

For our experiments, we use different MLP architectures with different depths (number of hidden layers). To distinguish different MLP architectures, the notation mlp^{n-h_1} (an input layer with n nodes and a hidden layer with h_1 nodes) or $mlp^{n-h_1-h_2}$ (an input layer with n nodes and two hidden layers with h_1 nodes in the first hidden layer and h_2 nodes in the second hidden layer) is used.

The feature vectors $\psi^C_{t\leq 1}(s)$ of the students have the length $|\psi^C_{t\leq 1}(s)| = 3$ with $\psi^C_{t\leq 1}(s) = (x_1, x_2, x_3) \in \{0, 1\}^3$ (where $x_i = 1$ if the exam c_i was passed and 0 else). A corresponding network of the form $mlp^{3-h_1-h_2}$ is visualized in Figure 3.7. The hidden layers have tanh as activation function and the output node has σ (logistic function) as activation function to be suitable for a binary classification problem. The used architecture is a fully connected feedforward network.

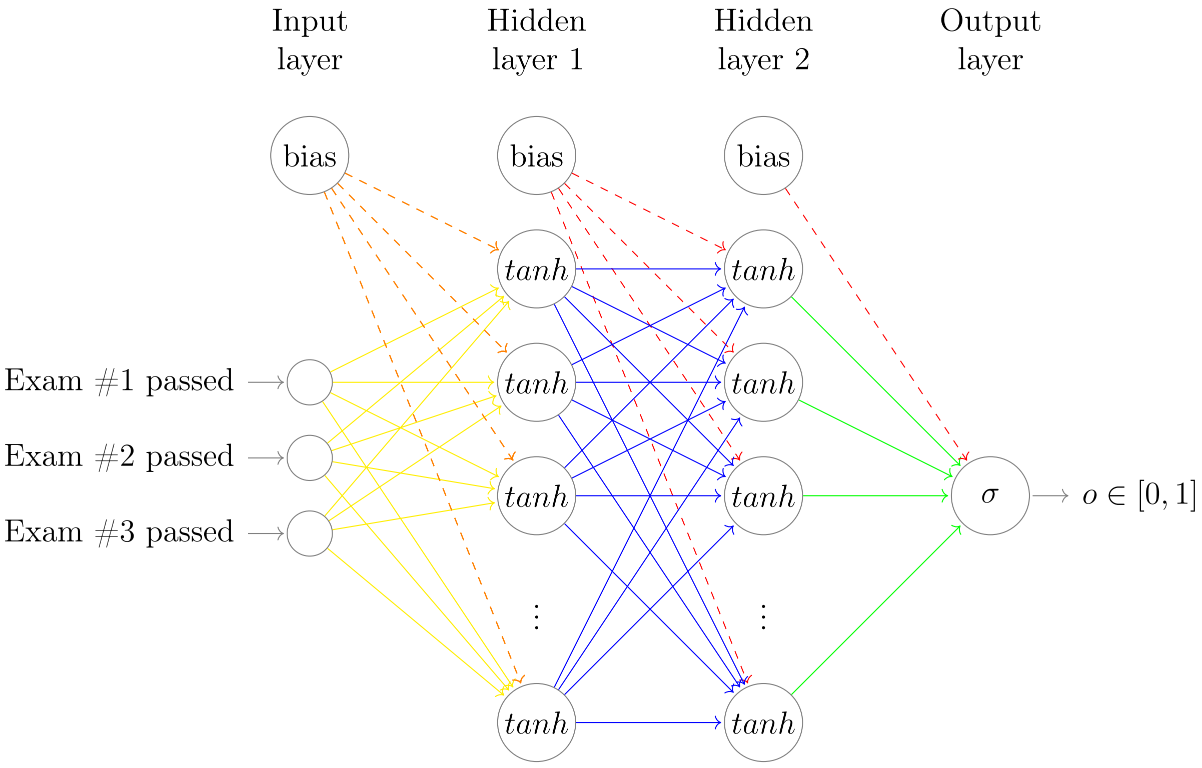


Figure 3.7: Our MLP model for dropout prediction after the first semester

Based on a case study for the case $k = 2$ semesters, we compared different architectures of MLPs (with one or two hidden layers) regarding different performance measures. For the evaluation of the different architectures (mlp^{12-5},mlp^{12-5-5}, $mlp^{12-5-20}$, mlp^{12-25}, $mlp^{12-25-5}$,$mlp^{12-25-20}$, mlp^{12-50}, $mlp^{12-50-5}$, $mlp^{12-50-20}$), we performed a stratified 5-fold cross validation for each architecture. Since the results in terms of recall, precision, F1 and AUC are almost equal (which means that all architectures are equally capable of adapting to the data), we choose the $mlp^{n-25-20}$ model (as it has a more balanced ratio of precision and recall) for further tests (where n depends on how many exams are used for coding the feature vectors in the respective tests).

3.4.4.2 SVM based Student Dropout Prediction

In the following experiments, we will examine both the time series-based representation of students and the non-time-series based representation of students. For the non-time-series based representation, we use the RBF kernel (Equation 2.8). Our SVM approach to handling time series data is presented below.

Time-series distances are necessary to calculate the distance between two students in the time-series representation. We used a distance based on multivariate *dynamic time warping* (DTW) (Ten Holt et al., 2007) and proposed a novel *weighted semester distance* (WSD) (Askinadze and Conrad, 2019) which is defined as follows:

$$d_{WSD}\left(T_k^C(s_1), T_k^C(s_2)\right) = \sum_{i=1}^{k} w_i \; d\left(\psi_{t\leq i}^C(s_1), \psi_{t\leq i}^C(s_2)\right)$$

When using binary feature vectors, the inner distance d of d_{WSD} is the same for the Hamming distance, Manhattan distance, and the squared euclidean distance.

Proof. For two binary feature vectors $x, y \in \{0,1\}^n$ the Hamming distance is defined by $d_{Hamming}(x,y) = \sum_i \mathbb{1}_{x_i \neq y_i}$. In the first part of the proof, we show $d_{hamming} = d_{manhattan}$:

$$\begin{aligned}
d_{hamming}(x,y) &= \sum_i \mathbb{1}_{x_i \neq y_i} && (3.1)\\
&= \sum_i^n \begin{cases} 0, & if\ x_i = y_i \\ 1, & if\ x_i \neq y_i \end{cases} && (3.2)\\
&= \sum_i^n \begin{cases} 0, & if\ (x_i = 0\ \&\ y_i = 0)\ or\ (x_i = 1\ \&\ y_i = 1) \\ 1, & if\ (x_i = 0\ \&\ y_i = 1)\ or\ (x_i = 1\ \&\ y_i = 0) \end{cases} && (3.3)\\
&= \sum_i^n \begin{cases} 0, & if\ |x_i - y_i| = 0 \\ 1, & if\ |x_i - y_i| = 1 \end{cases} && (3.4)\\
&= \sum_i^n |x_i - y_i| = d_{manhattan}(x,y) && (3.5)
\end{aligned}$$

In the second part, we show $d_{manhattan} = d_{squared-euclidean}$:

$$\begin{aligned}
d_{squared-euclidian}(x,y) = \sum_i^n (x_i - y_i)^2 &= \sum_i^n |x_i - y_i|^2 && (3.6)\\
&= \sum_i^n |x_i - y_i| = d_{manhattan}(x,y) && (3.7)
\end{aligned}$$

The equality between $|x_i - y_i|^2$ and $|x_i - y_i|$ follows from the fact that $|x_i - y_i| \in \{0,1\}$ for binary $x_i, y_i \in \{0,1\}$.

□

In Askinadze and Conrad (2019), we proposed to choose the weights in the following way:

$$w_i = \frac{i^2}{\sum_{j=1}^{k} j^2} = \frac{i^2}{\frac{1}{6}k(k+1)(2k+1)}$$

The denominator ensures that $\sum_{i=1}^{k} w_i = 1$. The idea of choosing the weights is based on the fact that the last known status of the exam results is the most important and, therefore, should be weighted the most. Exam results further in the past should have less influence on the distance. We have also tried other approaches to selecting weights. On the one hand, a weighting where the weights only increase linearly per semester $w_i = \frac{i}{\sum_i^k i}$. On the other hand, we tried a modified form of weighting, where additional weights per feature (exam result) were learned from the data (mutual information between feature and output). None of the additionally tested approaches could improve the results of the heuristic proposed above so that we will use these heuristic weights for further evaluation.

In order to use the SVM as a classifier based on time series data, we have proposed to adapt the RBF kernel by replacing the squared Euclidean distance (normally used in the RBF kernel) with the proposed WSD distance (which can use the Manhattan, Hamming or squared Euclidean distance equivalent as inner distance as shown above). This results in the following SVM kernel for two time series x and y:

$$K_{WSD}(x, y) = \exp(-\gamma \; d_{WSD}(x, y))$$

3.4.5 Evaluation

The prediction of study dropout is a binary classification problem. Many classifiers, including SVM, RF, and MLP, are able to provide probabilities instead of direct class assignments, so that for a student s to be classified, a value $f(s) \in [0, 1]$ is obtained. For the final assignment, a threshold θ can be used so that the final assignment is $F(s) \in \{0, 1\}$ is made as shown in Equation 3.8. For further attempts we choose $\theta = 0.5$. Smaller θ result in better recall values (and worse precision values) and larger θ result in better precision values (and smaller recall values).

$$F(s) = \begin{cases} 1, & \text{if } f(s) \geq \theta \\ 0, & \text{else} \end{cases} \tag{3.8}$$

For $k \in \{1, ..., 5\}$, we investigate how well the dropout can be predicted if only the exam data is available by the end of the k-th semester. For example, the training set for $k = 3$ includes only dropouts and graduates who have studied at least 3 semesters.

The results presented in Askinadze and Conrad (2019) are evaluated on a data set containing only dropouts who have registered for at least one exam in their studies. Since other research papers include all dropouts in their evaluation, we provide the results for all dropouts (including people who never registered for an exam) in order to compare the results with others better.

The number of exams $|C|$ used to create feature vectors depends on the study program (e.g., in some study programs, 3 and in others 10 exams per semester are intended). In our case study, we use a computer science program in which about 3

		Graduate			Dropout				Class sizes	
Sem.	Method	Pr	Re	F1	Pr	Re	F1	Acc.	Drop.	Grad.
1	MLP	0.79	0.65	0.7	0.82	0.89	0.85	0.8	752	424
	RF	0.73	0.71	0.71	0.84	0.85	0.84	0.8		
	RBF	0.8	0.63	0.69	0.81	0.9	0.85	0.8		
	WSD	-	-	-	-	-	-	-		
2	MLP	0.82	0.8	0.81	0.84	0.86	0.85	0.83	529	424
	RF	0.82	0.79	0.8	0.84	0.86	0.85	0.83		
	RBF	0.82	0.79	0.8	0.84	0.86	0.85	0.83		
	WSD	0.82	**0.82**	**0.82**	**0.85**	0.86	0.85	**0.84**		
3	MLP	0.84	0.88	0.86	0.87	0.83	0.85	0.86	404	421
	RF	0.83	0.87	0.85	0.86	0.82	0.84	0.84		
	RBF	0.86	0.86	0.86	0.85	0.86	0.85	0.86		
	WSD	**0.87**	0.88	**0.87**	0.87	0.86	**0.86**	**0.87**		
4	MLP	0.86	0.87	0.86	0.81	0.79	0.8	0.84	297	418
	RF	0.86	0.89	0.87	0.84	0.79	0.81	0.85		
	RBF	0.89	0.89	0.89	0.85	0.83	0.84	0.87		
	WSD	0.89	**0.91**	**0.9**	**0.87**	0.83	**0.85**	**0.88**		
5	MLP	0.9	0.9	0.9	0.84	0.82	0.83	0.87	246	413
	RF	0.89	0.92	0.91	0.87	0.81	0.83	0.88		
	RBF	0.9	0.92	0.91	0.87	0.83	0.85	0.89		
	WSD	0.9	**0.94**	**0.92**	**0.89**	0.83	**0.86**	0.89		

Table 3.1: Evaluation results based on a 3 times repeated stratified 10-fold cross-validation. All hyperparameters were determined on the training splits.

exams per semester are planned. Therefore, for the evaluation of the prediction after semester k, $|C| = 3k$ mostly taken exams until semester k are used.

The classifiers MLP, RF, and SVM (RBF Kernel) are not designed for time series. Therefore, for the representation of a student, the exam results at the end of individual semesters will be concatenated, so that a student s is represented by the vector $[\psi^C_{t\leq 1}(s) \oplus ... \oplus \psi^C_{t\leq k}(s)]$ of length $3k^2$ (concatenating k vectors of length $3k$) in the dropout analysis after semester k. For the WSD kernel the students are represented as the time series $T^C_k(s)$ with the shape $k \times 3k$ (k vectors of length $3k$). The results for the four approaches MLP, RF, SVM (RBF), and SVM (WSD) are shown in Table 3.1.

Discussion of the evaluation results

Since the group of dropouts who never took an exam always consists of vectors of 0s, it is easy for the classifier to classify this cohort of students as dropout correctly. This provides better evaluation results compared to Askinadze and Conrad (2019), where these students are not included in the evaluation.

The time-series approach only makes sense for the analysis of the prediction after at least 2 semesters, so that the WSD results are given from semester 2 onwards. The results show that the additional time information provided by the WSD kernel brings a small advantage in classification compared to non-time-series based classifiers. The results of WSD are in almost all cases at least as good and in most cases better than the other classifiers. Especially in the group dropouts, the precision value can be improved by 2 to 3% with the same recall value. In the group of graduates, the recall value can be improved by 2 to 3% with approximately the same precision value. The results show that in the first two semesters, it is easier to detect dropouts than graduates. From the 3rd semester on it is the other way round and recall values of the graduate group are clearly better than in the drop-out group so that from semester 4 on more than 90% of the graduates can be detected. The recall values of the dropouts become smaller with each semester because the easy to find group of students who never pass anything leave the university.

Comparing results with other studies

As already mentioned, it is difficult to compare the results of different studies because the data basis is often very different. A recent study (Berens et al., 2019) has also investigated dropout prediction for German students of two universities and has presented different classification results for both universities, so that the results strongly depend on the student body of the respective universities or even individual study programs.

Berens et al. (2019) have, among other things, investigated a feature set, which is also based on student achievement data only. The authors have used the following data to represent the feature vectors: “average semester grade, average semester credit points earned, number of registered but unattended exams, and the number of attempted but failed exams, number of most important exams passed in a given semester”. This feature coding can be classified as GFB since it calculates sums and averages of local information. For the evaluation, two different strategies were used to choose θ. As described at the beginning of this chapter, the θ parameter determines whether the result has a good recall value or a good precision value. In the first attempt, θ was chosen so that Precision=Recall. Using the AdaBoost classifier, the following results were achieved:

- after the 1st semester (Re=73.83%, Pr=73.83%, Acc=78.53%)
- after the 2nd semester (Re=74.95%, Pr=74.95%, Acc=82.43%)
- after the 3rd semester (Re=80.58%, Pr=80.58%, Acc=87.62%)
- after the 4th semester (Re=79.94%, Pr=79.94%, Acc=89.63%)

In the second case, θ is the average dropout rate. Since the dropout rate is below 0.5, a higher recall value is usually achieved, but the precision value drops. Therefore, for the prediction after the 4th semester, the recall value 92.55% was reached. The authors have not given a precision value, but this can be calculated from the numbers of the confusion matrix and is for the 4th semester: 60.26%.

How to choose θ? That depends on what is important in predicting student dropout. If we want to find as many dropouts as possible and it is not important if students who are not at risk are mistakenly classified as endangered, we should choose a smaller θ. If we want to be sure that found at-risk students are actually at risk, θ should be increased. Aulck et al. (2019) suggested to reduce the false positives (even if false negatives increase) when developing an alert system.

Findings: We can learn from the results:

- Adding additional time-series information of the exam results can give slightly better classification results at higher semesters.
- The information whether an exam has been passed is sufficient to produce similarly good results as when using significantly more data.
- The model in Berens et al. (2019) was trained on the exam results of 12,730 students. Our evaluation shows that a much smaller number (approx. 500-1000) of training elements is sufficient to generalize similarly. This means that universities or study programs with a small number of students are able to make useful predictions.

Analysis of misclassifications

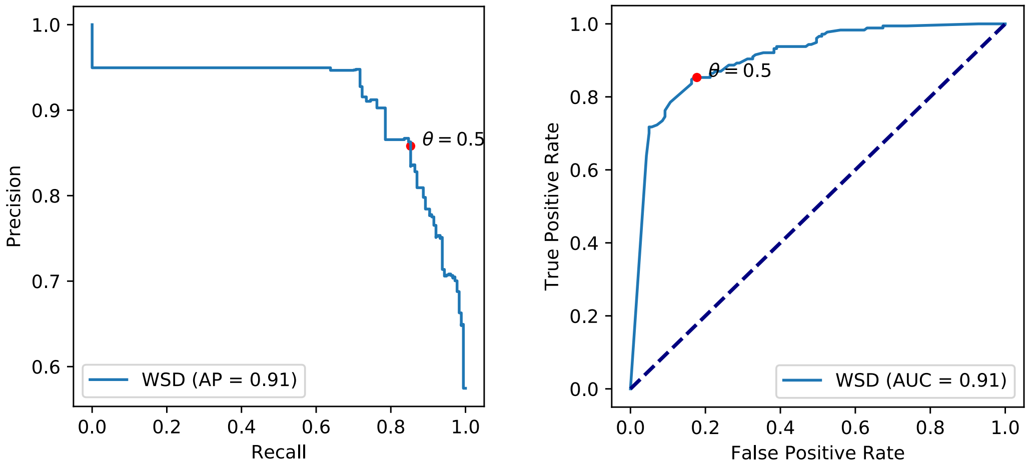
We now explore for the WSD approach, which time-series patterns cause misclassification. For this, we consider the case of the prediction at the end of the 2nd semester. 66.6% of the students (635) are used for training and 33.3% for testing (318). The allocation is made so that the proportion of dropouts and graduates is the same in both splits. The results (recall, precision and F1) for both classes graduates and dropouts are shown in Table 3.2 and the confusion matrix in Figure 3.9a. The results correspond to the cross-validation results for the 2nd semester in Table 3.1 with small deviations.

Class Name	precision	recall	F1-score	support
Graduates	0.82	0.82	0.82	141
Dropouts	0.86	0.85	0.86	177

Table 3.2: Prediction results (recall, precision, F1)

The results are additionally visualized as ROC and recall-precision curve in Figure 3.8. In Figure 3.8a, we see that with the threshold $\theta = 0.5$ the recall value is almost equal to the precision value. If we increase θ, a precision value of over 90% could be achieved (when recall falls to about 65% accordingly). In the ROC plot (Figure 3.8b), we see that an AUC value of 0.91 has been reached. To more clearly demonstrate the effect of θ, the measures precision, recall, and F1 are visualized in Figure 3.9b. It becomes clear that the values of recall and precision can be manipulated with the θ parameter and that F1 is not a good measure, because F1 hardly changes for $0.4 \leq \theta \leq 0.6$, while recall and precision have strong changes in this interval.

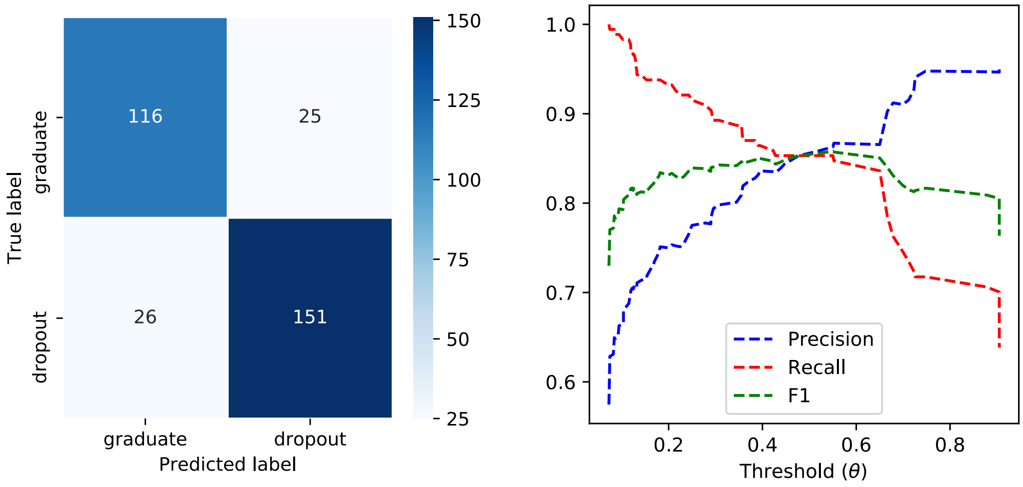
We take a closer look at the misclassified students: The wrongly classified students are grouped according to the same exam result after the first and the second semester. The groups with at least 3 elements are displayed in descending order based on group size in Table 3.3. These 3 groups already represent about 33.3% of all miss-classified students. From the table, it becomes clear that the wrong classifications often occur when students who have passed only a small number of exams graduate later in their studies and vice versa. The exam patterns of the first two groups with the most miss-classified students were classified as dropout though they actually belong to graduates (false positives). The group "(100000), (100100)" is the most often misclassified group



(a) Recall-Precision curve

(b) ROC

Figure 3.8: Prediction results (recall-precision and ROC curve)



(a) Confusion matrix

(b) Threshold curve

Figure 3.9: Prediction results (confusion matrix and threshold plot)

with a frequency of 8 (15.69%). This pattern occurred 37 times in the train split, 24 times as dropout, and 13 times as graduate. So the decision of the SVM here to classify this pattern as dropout (which is the majority class in the training split) is correct. The same is also true for the second most common falsely classified pattern. These are the students who did not pass any exam. Again, the decision to classify them as dropout is correct in principle. This observation shows us that it is not possible to achieve a perfect classification (100% recall and precision), because there is no unique assignment $f : exam\ results\ pattern \rightarrow \{0, 1\}$ (f is not a function in the mathematical sense because each element of the domain of f may be mapped to more than one element in the target set). More features are needed to better separate two students with the same exam results but different outcomes (graduate and dropout). In particular, the reasons for dropping out can be very different, as already mentioned, so that the dropout cannot always be predicted from the exam results.

exam results	incorrectly classified as	count	% of mis-classified	as dropout in train split	as graduate in train split
(1 0 0 0 0 0), (1 0 0 1 0 0)	dropout	8	15.69	24	13
(0 0 0 0 0 0), (0 0 0 0 0 0)	dropout	6	11.76	220	17
(1 0 0 0 0 0), (1 0 1 1 0 0)	graduate	3	5.88	0	6
		17	33.33	244	36

Table 3.3: Analysis of most miss-classified time series exam result patterns

3.5 Student Performance Prediction

Student performance prediction is most often about predictions at the course level. The goals of prediction are usually the prediction of grades and the course pass/fail prediction. Since pass or fail in a course often depends on the grade, this prediction is often a by-product of the grade prediction. Shahiri et al. (2015) examined the features used to predict grades and found that the important features include previous achievements (mostly cumulative grade point average and features based on quizzes, lab work, and attendance) and demographic attributes. Studies such as Al-Radaideh et al. (2006) and Cortez and Silva (2008) have shown that using only demographic features does not yield promising results. Al-Radaideh et al. (2006) used various demographic features and the highschool grade with decision trees to predict the final grade in a course. The most important attribute was the only non-demographic attribute highschool grade due to the highest gain ratio in the root node. However, with this feature set, they were only able to achieve accuracy values below 40%.

Cortez and Silva (2008) investigated how well the final grade can be predicted based on a feature set with only demographic data and a feature set that also contained previous grades. The results showed that if only the demographic features were used, the results would be about the same or slightly better than if a naive classifier was used which always selects the majority class. However, as soon as earlier grades were added, the results were significantly improved. Furthermore, most demographic features were irrelevant for their prediction.

Due to the new available digital e-learning services, such as web-based training, students can learn interactively and their usage behavior can be tracked. In Chapter 3.3, we explained the xAPI architecture, which allows to track and integrate data from heterogeneous modern digital learning environments. Features that are created from such data are often called e-learning/behavioral features. In a recent study, Saa et al. (2019) examined the most important features for predicting student performance and, like Shahiri et al. (2015), found that past performance and demographic features are among the most important but added the e-learning behavioral features to the top 3 most important features.

In Chapter 3.5.1, we investigate how well grade levels can be predicted based on a mix of demographic data and aggregated xAPI (behavioral features) data. In Chapter 3.5.2, we investigate the prediction of scores based on raw xAPI data collected during the learning with an eBook.

3.5.1 Demographic and Behavioral Data

Amrieh et al. (2016) tried to predict the students' grade level (low, middle, high) at course level on a data set[5] with 16 features. For the e-learning behavior data in this data set, a tracker was used to collect usage data in xAPI format in an LMS. The xAPI statements were aggregated to extract four features describing the usage intensity of the LMS: *participation in discussion groups, visited resources, raised hand on class, and viewing announcements.* In addition, the features used include the academic background and demographic features. The authors investigated how well the prediction works with and without behavioral features. If behavioral features were added to the remaining features, the accuracy result for this multi-class problem could increase from 55.6% to 75.8% using decision trees. This shows that LMS behavioral features are relevant.

In Chapter 3.4, we discussed that feature sets that consist of previous exam results give almost as good results as feature sets that include additional demographic information, showing that demographic information hardly contributes any additional relevant information. To investigate whether it is similar for behavioral features that demographic features do not significantly improve the final result, we test the prediction only for behavioral features, which was not investigated by Amrieh et al. (2016). We performed a 5-fold cross-validation with the random forest classifier on the data set and were able to achieve an accuracy value of 64.88% only using the behavioral features. This result shows that behavioral features alone are not sufficient to make the best possible predictions, so adding demographic features is necessary.

The number of features can grow rapidly, so that the number of features must be reduced both to improve the model and to make it easier to interpret. As mentioned in Chapter 3.1.4, finding an optimal minimal feature set is an NP-hard problem, so greedy filter-based feature selection approaches are mostly used, which sort the features by a quality criterion and select the best k features. For the feature selection, Amrieh et al. (2016) chose an information gain (mutual information) based approach and selected 10 features that achieve the above-mentioned accuracy result of 75.8%.

In the following, we investigate whether other feature selection methods are more suitable or can produce the same results with even fewer features. Since the data set consists of several categorical features, several preprocessing steps are necessary. Categorical features with two different possible values were binary coded and features with more than two categories were transformed with one-hot-encoding, resulting in a data set with 66 features.

As an alternative to mutual-information-based feature selection, we investigate the use of ANOVA (Chapter 3.1.4.1.4). The 66 features are sorted by their F-value. Then, we examine how the subsets, including only the best feature, the best two, best three, etc. affect the classification result. Figure 3.10 shows the 3 times repeated stratified 10-fold cross-validation results (a total of 480 students, i.e., 432 in the training and 48 in the test set in each run) for the feature sets containing up to 15 best features out of a total of 66 features for both methods *mutual information* and *ANOVA*. The remaining 51 are not included in the visualization for reasons of space, and as they do not contribute to a higher classification result. After the 14 best features were used, there is no significant improvement in the classification results anymore. According to ANOVA, the best 14 features are as follows:

[5] `https://www.kaggle.com/aljarah/xAPI-Edu-Data`

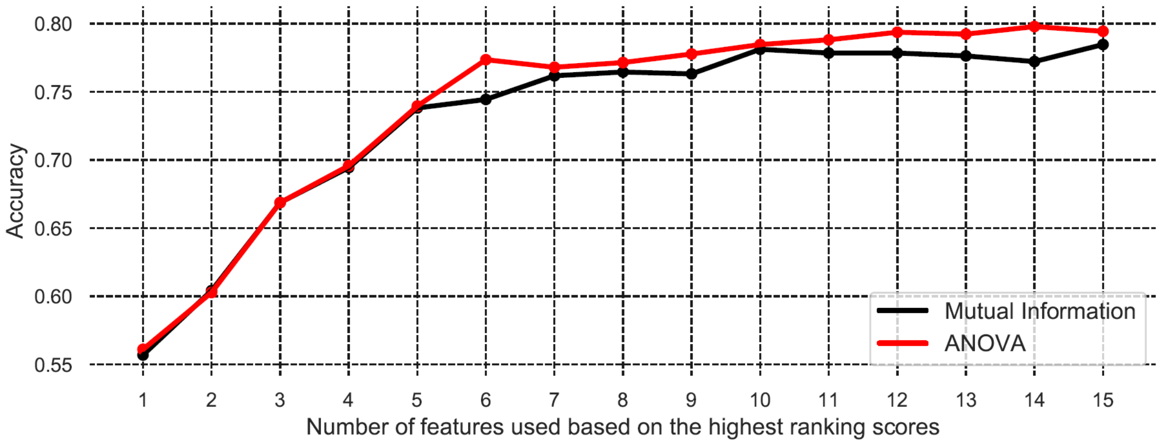
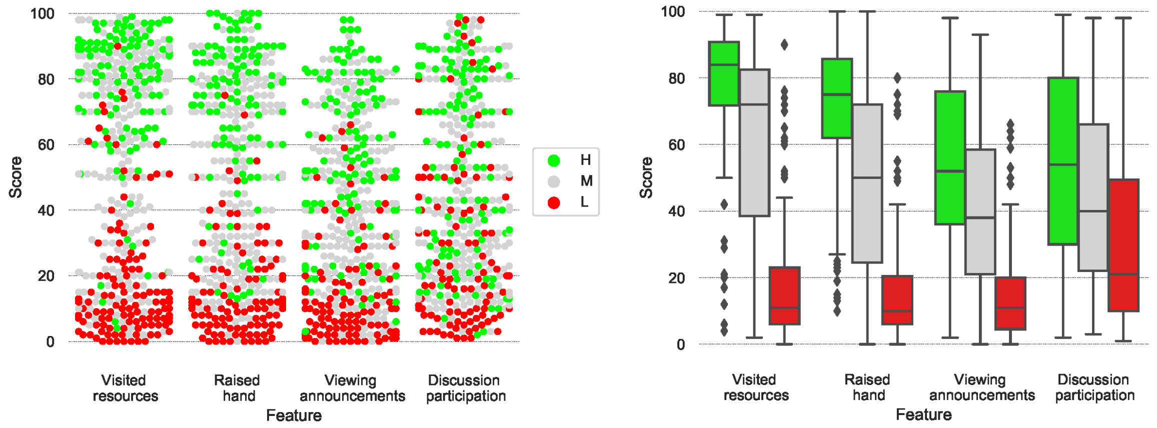


Figure 3.10: Accuracy results for an increasing number of features with the best F-values

1. **Visited resources**
2. Student absence days
3. **Raised hand**
4. **Announcement Views**
5. Parent answering survey
6. Responsible parent
7. Parent school satisfaction
8. **Discussion**
9. Gender
10. Place of birth: Kuwait
11. Nationality: Kuwait
12. Place of birth: Libya
13. Nationality: Libya
14. Course topic: IT

The behavioral features are shown in bold and all of them are included in the top 8 features. Since a different feature selection method was used here than in Amrieh et al. (2016), the features found differ partially. These best 10 features achieve an accuracy value of 78.47%, which is approximately 3% better than the RF result (75.6%) in Amrieh et al. (2016). Figure 3.10 shows that the feature sets of the two examined filter-based univariate feature selection methods differ from each other and that the ANOVA feature sets always deliver at least as good results as the mutual-information-based feature sets. The top 6 features (according to ANOVA) achieve an accuracy value of 77.57%, which can only be exceeded by adding another 5 features. This is particularly interesting because, with the 6 features found by this approach, we can avoid using the ethically questionable features gender, place of birth, and nationality without losing much in terms of accuracy. This is also good since the data minimization required by GDPR Art. 5.1 is achieved by this selection of features.

Using only the 4 behavioral features achieves an accuracy value of 64.55%. All four behavioral features are included in the top 8. In Figure 3.11, the four behavioral features are visualized in a swarm plot and a box plot. The swarm plot reveals some detailed information about the distribution of the individual students (each student is represented by a point), while the boxplot shows us the average values and quartiles more clearly. The colors (green, gray, and red) symbolize the three cohorts/classes (based on the final grades) H=high, M=middle, L=low. The mean values μ_1, μ_2, and μ_3 of the different classes are unequal for all four features, so the null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3$ (which is examined by ANOVA) can be rejected. That shows us why ANOVA considered these four features as relevant. The fact that the data of the individual populations is not normally distributed is not a problem due to the robustness of ANOVA (Blanca et al., 2017). We can see from the plots that the majority of students belonging to class L is less likely to open the resources available in the LMS. These students are also less likely to be active in the classroom and are less likely to look at the announcements than students of the other two cohorts (M and H). Conversely, the majority of students belonging to the cohort H are more active in both LMS and the classroom than the other two cohorts. This analysis shows that behavioral features are useful for predicting student grades and, especially in combination with other feature types, provide useful results.



(a) Swarm plot

(b) Boxplot

Figure 3.11: Behavioral features

As already mentioned, data protection issues are important in the EDM area. For privacy reasons, some students may not allow certain information about them to be stored, resulting in records with missing values (Askinadze and Conrad, 2018b). Therefore, it should be examined how missing values in the features affect the ML models already during the selection of the features. Additionally, we investigate how missing values in these 6 "most" relevant features affect the classification result. For this purpose, we simulate missing values in each of the 6 features and replace them with an imputation strategy. The advantage of imputation is that even if some feature values are missing in the data of some students, their data can still be used to train a model. Otherwise, their data would not be useful for the creation of a model.

As imputation strategy, we choose 10-nn (10 nearest neighbors), which means that the missing values of a feature are calculated by the average of 10 feature values, whereby the 10 neighbors are found with respect to the remaining 9 features (whose values are known).

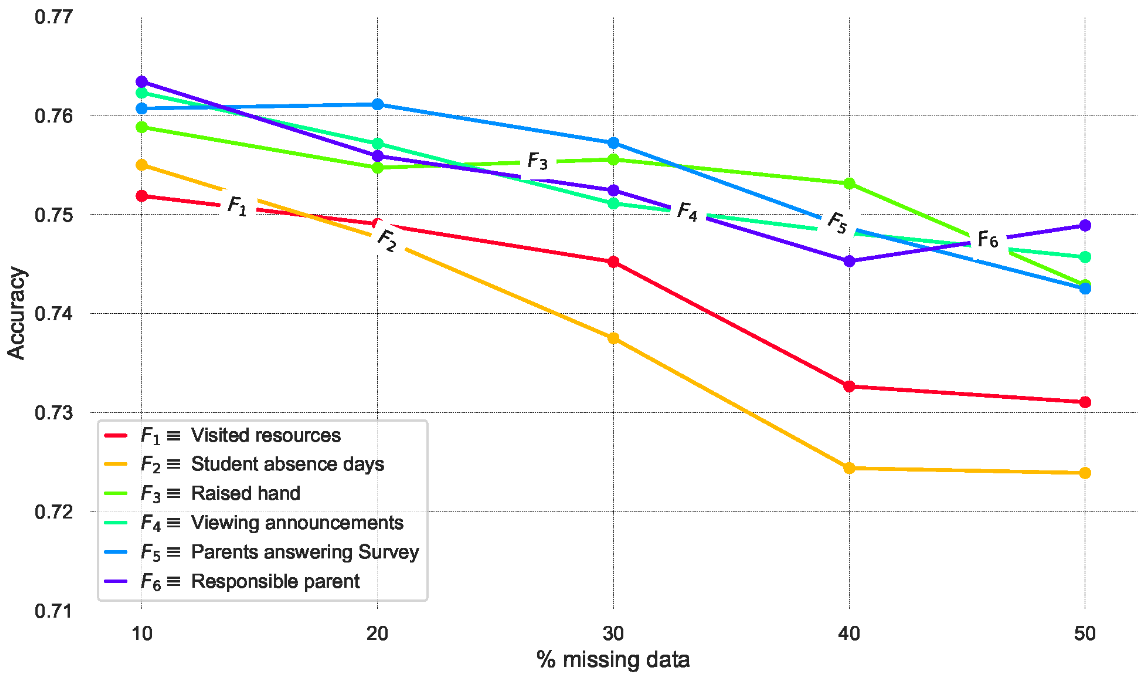


Figure 3.12: Investigation of the influence of missing values in individual features

As stated in Chapter 3.1.4, strongly relevant features are those that contribute information that is not contained in others and weekly relevant features are those that carry information that is also contained in other features. Using the 10-nn missing value imputation strategy, one feature is predicted by the rest. This means that strongly relevant features probably cannot be predicted well from other features, while weekly relevant features can be predicted better.

The results (based on a 3 times repeated stratified 10-fold cross-validation) are visualized in Figure 3.12 (a total of 480 students, i.e., 432 in the training and 48 in the test set in each run). The effects of missing values vary per feature. Even a high percentage of missing values for some features does not lead to a strong loss of accuracy. In contrast, the both most important features *visited resources* and *student absence days* (according to ANOVA) are more sensitive to missing values. The differences between the features regarding missing values are probably because some features are more relevant than others, i.e., carry information that cannot be predicted from other features. However, even with 50% missing values for relevant features, the accuracy decreases by only about 5%. This effect is increased if missing values can occur in more than one feature. For 50% missing values in all six features, the accuracy value drops to 62%. In Askinadze and Conrad (2018b), we have examined a similar grade-level prediction problem on another data set, and with 50% missing values, the accuracy has also decreased by approx. 20%.

3.5.2 Behavioral Clickstream Data

In this chapter, we examine a particular case of performance prediction, where only e-learning behavioral data is available. More specifically, we do not investigate LMS behavior (such as the number of logins or number of announcement views as in Chapter 3.5.1), but behavior data (xAPI statements) with interactive learning tools such as web-based training or eBooks.

Prediction based on raw xAPI interaction data is a current research topic that has been addressed in recent workshops *LA@ICCE2018: Joint Activity on predicting student performance* (Flanagan et al., 2018) and *Data Challenge: Predicting Performance Based on the Analysis of Reading Behavior* (Flanagan et al., 2019). In both workshops, raw xAPI data of eBook reading behavior was provided to different participating research teams. In addition to the pure interaction data, a student's final grade was given, based on a test on the contents of the eBooks. The participants of the two workshops have tried to solve different prediction problems. On the one hand, the prediction whether a student belongs to the *high performers* (final grade belongs to the best 50%) or *low performers* (final grade belongs to the lower 50%). On the other hand, they tried to predict the exact grade (regression problem). Examining whether or not the test was passed did not make much sense, as the test was only failed in a few exceptional cases.

The final grade was given on a scale between 0 and 100 points. With a balanced (50/50) separation of the low performers and high performers, the point threshold for the Data Challenge (LAK 2019) is about 85 points. Jihed and Mine (2019) were only able to achieve an accuracy value of 53% and suggested to lower the point threshold to 80 so that the two classes were no longer balanced. The accuracy value then improved to 68%. Hirokawa and Yin (2019) have also worked with the 80 point threshold and reached an accuracy value of about 92% with their approach.

To find features that are well suited to distinguish low performers from high performers, we applied t-tests in Askinadze et al. (2018) and extracted some interesting features based on the xAPI statements *operationname_PREV*, *xAPI_read*, *operationname_ADD_MARKER*, and *operationname_ADD_MEMO*. These features were chosen because students who belong to the group of high performers, on average, read more pages in the eBook, click back more often, save more memos, or mark texts. In a recent study by Akçapınar et al. (2019), the authors also used t-tests on raw eBook xAPI data to find differences between these two groups of students.

With our two contributions Askinadze et al. (2018) and Askinadze et al. (2019a), to the workshops/challenges, we provided two approaches to the exact grade prediction based on behavioral clickstream data. In both cases, the RMSE results were about as good as a self-made baseline (average of all grades). Other teams, such as Lu et al. (2018), were also unable to provide RMSE results that differed significantly from the baseline. Ng et al. (2019) also found that the results of their deep learning approach are about as good as the baseline, and in their conclusion, they mentioned that current ML techniques cannot yet efficiently solve the given regression problem. For the upcoming LAK20 Data Challenge[6], the regression problem based on the reading behavior clickstream data was set as the main task.

[6] https://sites.google.com/view/lak20datachallenge/home

4

Educational Dashboards

"Visualization gives you answers to questions you didn't know you had."
— Ben Schneiderman (Kirk, 2012)

In his introduction to *performance indicators* (PI), Fitz-Gibbon (1990) mentioned that those responsible for managing complex systems (e.g., in the education sector) need key indicators to support them. He defined a PI as "an item of information collected at regular intervals to track the performance of a system". The author pointed out that PI may not be ideal but are important for the quality control of a system.

Several helpful indicators may be displayed together in a system (e.g., a dashboard). Few (2004) defined a dashboard as "a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance". He also noted that a dashboard is not about presenting a certain kind of information (like PI), but about how the information is presented to achieve a specific goal.

Dashboards used in the educational context are often called educational dashboards, learning dashboards, or learning analytics dashboards (LAD). Bajzek et al. (2007) gave one of the first found definitions for such dashboards: "a tool that provides visibility into key indicators of student learning through simple visual graphics such as gauges, charts, and tables within a web browser". Schwendimann et al. (2016) defined a learning dashboard as "a single display that aggregates multiple visualizations of different indicators about learner(s), learning process(es) and/or learning context(s)".

In Verbert et al. (2014), an analysis of existing 24 learning dashboards was carried out. The authors divided the examined dashboards into the following three categories:

1. "Dashboards that support traditional face-to-face lectures": The goal of these dashboards is to help the tutors by giving them live feedback from their students.

2. "Dashboards that support traditional face-to-face group work": The goal of these dashboards is to help tutors manage group work by, e.g., getting an overview of each group's activities.

3. "Dashboards that support awareness, reflection, sense-making, and behavior change in online or blended learning": These dashboards can support both teachers and students in blended learning (face-to-face teaching combined with e-learning).

Millecamp et al. (2019) mentioned that learning dashboards are usually described using only stakeholders and objectives. For the dimension *objective*, the authors give the two sub-dimensions *reflective* and *predictive* and named the following four sub-dimensions for the dimension "stakeholder": *institutions*, *learners*, *teachers*, and others. Klerkx et al. (2017) listed objectives of various dashboards: "providing feedback on learning activities, supporting reflection and decision making, increasing engagement and motivation, and reducing dropout". Büching et al. (2019) stated that the various stakeholders (students, lecturers, university administrators, or governmental institutions) have interests at different levels: micro-level (courses), meso-level (courses) and macro-level (universities).

In parallel to learning analytics (using data from learning management systems), terms such as institutional analytics (using institutional student data) and academic analytics (using data from the student information system and guidance system) can also be used (Elias, 2011; Daniel, 2015; Büching et al., 2019) to differentiate which data is used and what the objectives of the analysis of this data are.

Schwendimann et al. (2016) examined a larger number of articles on learning dashboards (55 out of 346 crawled articles were selected according to their quality criteria). They found that the largest target group of users of dashboards studied were teachers (75%) and students (51%). Administrators/institutions were rarely found as a target group. In addition, they found that the dashboards had three main goals: self-monitoring (51%), monitoring others (71%), and administrative monitoring (2%). These results show that much research is being done on teachers' and students' dashboards, but little research has been done on developing administrative dashboards.

The remainder of the chapter is structured as follows: In Chapter 4.1, we present the development of an administrative dashboard based on the example of a German university. In Chapter 4.2, we give an overview of early warning systems, a type of dashboards based on student dropout and performance prediction. Finally, in Chapter 4.3, we discuss data protection issues.

4.1 Administrative Educational Dashboards

Daniel (2015) explained that academic analytics will play an important role for the administration of universities in the future, as it enables data analysis on an institutional level (e.g., an overview of what happens in an entire study program) and not like learning analytics, which focuses on the learning process itself. Objectives of academic analytics is thus, among other things, the support of the administration or the persons who are responsible for the strategic planning of the educational institutions. Institutional analytics has a similar role, the goal of which, according to Daniel (2015), is, among other things, the use of dashboards to support the decisions of all departments and divisions of the institution.

Since the central university administration often does not have the data of different e-learning systems of individual lectures and courses, the only available data is the enrollment information and the study progress data of the students (this data is only added if a student registers for an examination or takes part in an examination). Even if the administration only has the study progress data at its disposal, these can be used to identify at-risk students as shown in Chapter 3.4 or gain insight into the study behavior of the students. In Chapter 4.1.1, we use the example of a German university and show first visualization attempts to gain information from this limited data.

4.1.1 German University Data

While there are already several examples for the use of Learning Analytics in the Anglo-American area, the development in German universities is still at the very beginning (Sclater et al., 2016; Büching et al., 2019; Hartl, 2019).

For this study, we were provided with the data of an computer science (CS) study course of a German university. We have found that the data available to universities is often limited to a small amount of information required for enrolment. The following data on students is available in anonymous form:

- Matriculation number: An anonymous number assigned to the student.
- Enrolment semester: The semester in which the student enrolled at the university.
- Graduated: The information on whether the student has completed the course of study.
- Exmatriculated: Information whether the student has been exmatriculated from the university.
- Still studying: Information whether the student is still studying the subject in which he started his studies at the university.
- Subject changer: Information if a student has changed his study subject.

Mostly at the end of the semester, exams are taken, and exam data of the students is added to the central administration system. This data includes the following information:

- Exam status: The information indicates whether a student registered for an exam or whether the student passed or failed the exam.
- Exam regulations: The examination regulations provide a framework that describes which examinations have to be taken and how they have to be taken in order to complete a course of study successfully. The examination regulations can have different versions.
- Exam semester: The semester in which the exam attempt was made.
- Exam title: The title of the module to which the respective exam belongs.
- Exam recognition: Information indicating whether the exam is a recognized exam.
- Exam attempt: This information indicates in which attempt the student is in the respective exam.

- Exam grade: This value indicates the grade obtained in the exam (if it is a graded exam).

- Exam credit points (CP): This value specifies how many CP are assigned to the module, which is to be examined.

Even this limited amount of data can be used to get a first insight into the study behavior. In Table 4.1, we present the number of matriculated students of individual cohorts who started their study in different winter semesters (WS) from 2002 to 2011. The data examined is from the year 2016, so that all students had at least the standard period of study (6 semesters) to graduate. We indicate the quantity of all enrolled students in column “enrolled”. With the help of the information about the exam registrations, we have found that a large number of students (approx. 40%) do not have a single exam registration during their studies. For further visualizations and studies, we only use the students who had at least one exam registration. The group of students who registered for at least one exam attempt is divided into the four sub-groups “graduated”, “still in CS study”, “subject changers still in study”, and “dropouts”. The percentages given for these 4 sub-groups refer to the proportion of students with at least one exam attempt. On average, about 43% of these students completed their studies with a degree. A further observation is that about 44% of students who registered at least one exam attempt in CS dropped out. The remaining 13% of the students are still enrolled either in CS or another subject at the university to which they have changed.

Table 4.1: Overview of the data of computer science students from different cohorts

cohort	enrolled	students with at least one exam attempt in CS									
		total		graduated		still in CS study		dropouts		active subject changers	
WS	∑	∑	%	#	%	#	%	#	%	#	%
02/03	184	107	58,2	45	42,1	3	2,8	57	53,3	2	1,9
03/04	243	113	46,5	70	61,9	6	5,3	36	31,9	1	0,9
04/05	130	84	64,6	39	46,4	2	2,4	42	50	1	1,2
05/06	185	127	68,6	56	44,1	4	3,1	66	52	1	0,8
06/07	143	106	74,1	51	48,1	7	6,6	48	45,3	0	0
07/08	116	80	68,9	35	43,8	5	6,3	39	48,8	1	1,3
08/09	110	65	59,1	28	43,1	5	7,7	28	43,1	4	6,2
09/10	113	73	64,6	28	38,4	13	17,8	31	42,5	1	1,4
10/11	135	86	63,7	28	32,6	14	16,3	34	39,5	10	11,6
11/12	293	143	48,8	44	30,8	36	25,2	55	38,5	8	5,6
∑	1652	984	59,6	424	43,1	95	9,7	436	44,3	29	2,9

The tabular overview provides an initial insight into various performance indicators, but this presentation does not reveal the study behavior of individual students.

Hörnstein et al. (2016) suggested the use of CP (and their derivatives) as performance indicators for study program monitoring, as CP are suitable for measuring study success. The authors suggested different visualizations of study success based on CP and bar charts.

In addition, we propose the use of heat maps to visualize the CP achieved by individual students per semester. Figure 4.1 shows, for example, the CP achieved per semester by graduates of the WS 02/03 cohort. The different colors symbolize the number of CP reached. The color, which corresponds to 40 CP, means 40 CP and more. Each row visualizes a student, so that it becomes clear how many CP the individual students received in which semester. The students are sorted according to the duration of their studies so that the students who have studied the longest are shown above.

At least 180 CP are required to complete the Bachelor's program. A standard period of study of about 6 semesters corresponds to about 30 CP per semester. Looking at the graduates of the WS 02/03 cohort, the heatmap shows that they can be roughly divided into two groups. Students who achieve their degree in 6-7 semesters are able to receive 30 CP in the first semester. Students who achieve less CP in the first semester tend to need longer for a degree.

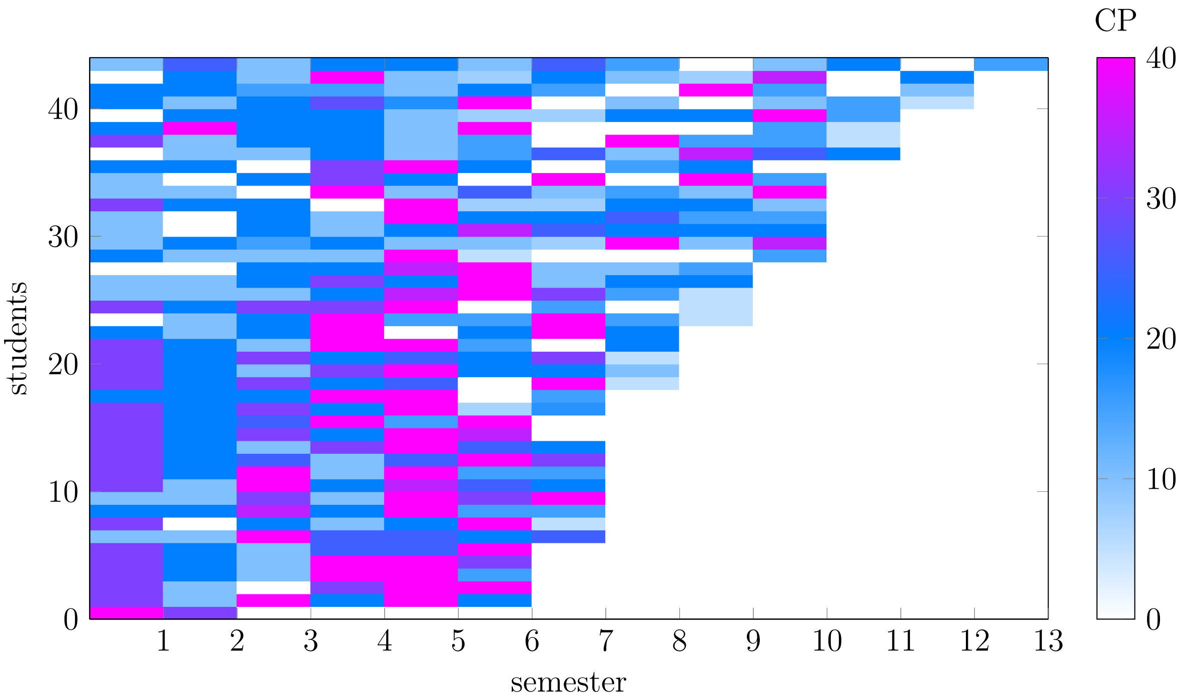
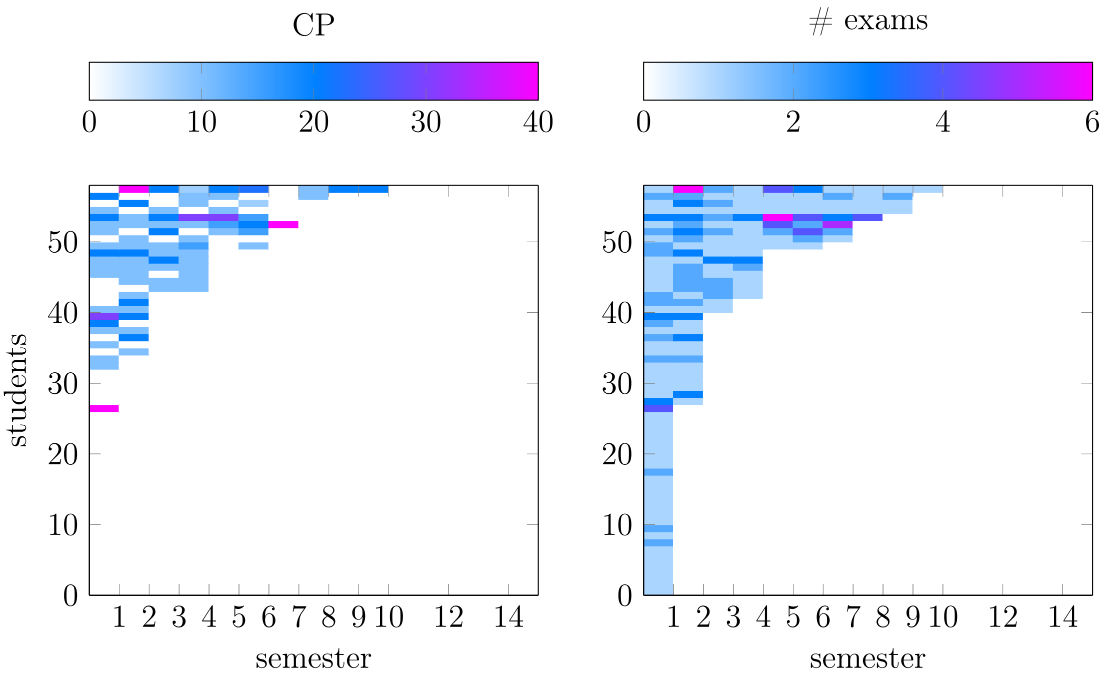


Figure 4.1: CP per semester of graduates of WS 02/03 Cohort

As a further indicator for the use of heat maps, we recommend the number of exams a student takes per semester. In Figure 4.2, we visualize the students of the WS 02/03 cohort who completed their computer science studies without a degree. In each line, Figure 4.2b shows the progress of a student who dropped out, and the color symbolizes the number of exams taken in each semester. This visualization shows that students who prematurely leave the university often only have one or two exam registrations per semester.

Using the example of heat maps, we shown how a deeper insight into the study progress data can be given with the help of visualizations. The manual creation of such heat maps and other advanced visualizations is very time-consuming, which requires a dynamic solution for filtering and visualizing the data. In the next subsection, we describe the development and challenges of a dashboard capable of creating such visualizations automatically for selected cohorts.



(a) CP per semester

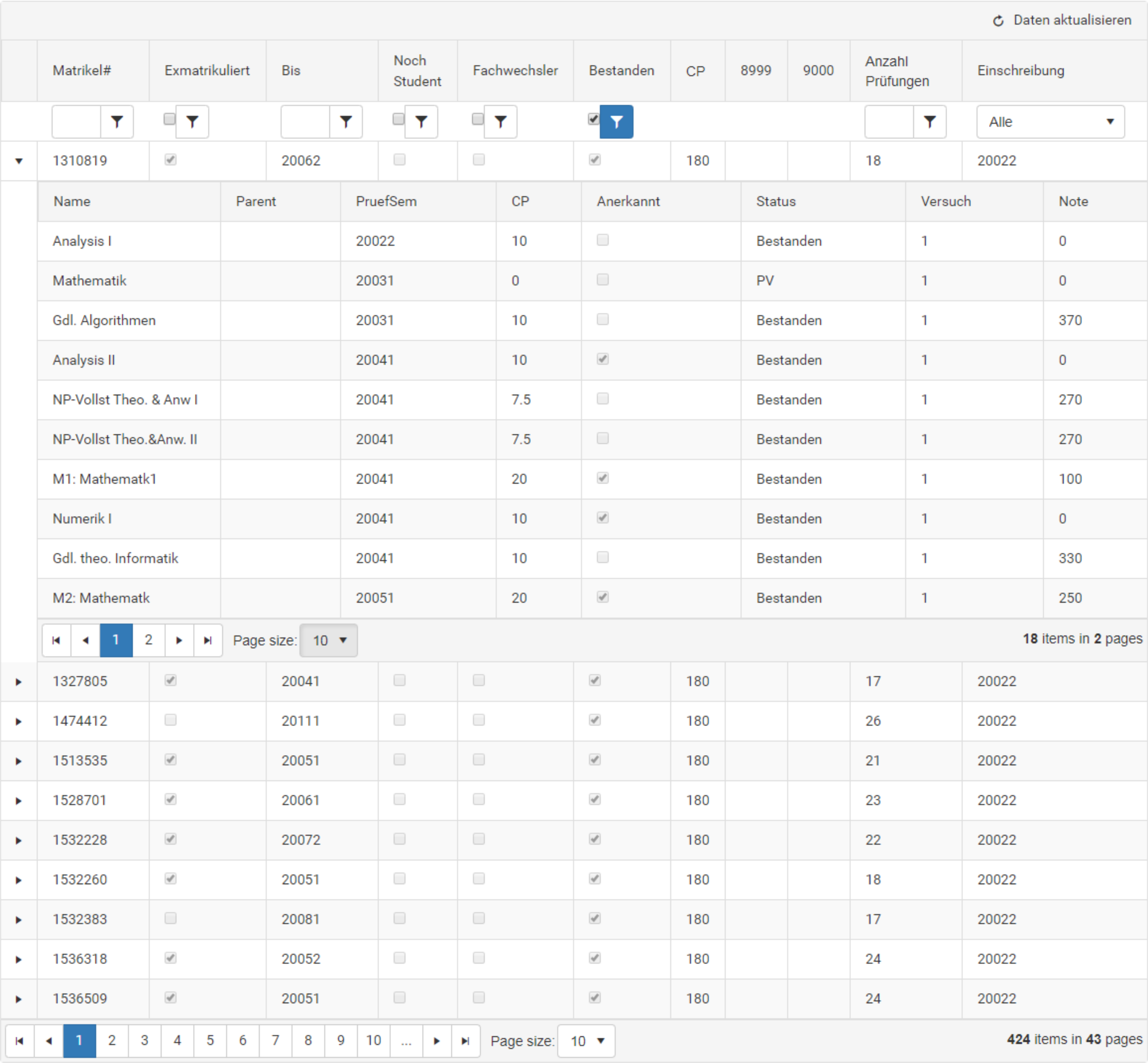
(b) Examinations per semester

Figure 4.2: Students of the enrolment semester WS 02/03 with at least one examination who have not completed their CS studies.

4.1.2 Development of a Study Progress Monitoring Dashboard

In order to develop our dashboard, the integration of student data must first be enabled. If a dashboard should not be adapted to the data structure of a single university but should allow the integration of data from different universities, possible deviations in data storage must be considered. In addition to the data set presented in Chapter 4.1.1, we were also provided with the data set of another computer science course at another German state university. Although the new data is very similar, the exams and modules are hierarchically structured in contrast to the data already described. In Askinadze and Conrad (2018a), we described the simplified data model to integrate the data of both hierarchically and non-hierarchically structured modules of a study program. To import the data into the dashboard, a university must simply export its own data in one of the two given formats (hierarchical or non-hierarchical). The dashboard provides two parsers for both types of data to import and store in its own database.

Once the data is integrated into the system, all the advantages of a dashboard such as browsing, filtering, and predefined visualizations can be used. Figure 4.3 shows the view of the dashboard, which allows fast filtering of student data on different parameters and for different cohorts. In the following, we show examples of various visualizations we proposed and implemented that can be used in an administrative dashboard.



Daten aktualisieren

	Matrikel#	Exmatrikuliert	Bis	Noch Student	Fachwechsler	Bestanden	CP	8999	9000	Anzahl Prüfungen	Einschreibung
		□		□	□	☑					Alle
▾	1310819	☑	20062	□	□	☑	180			18	20022

Name	Parent	PruefSem	CP	Anerkannt	Status	Versuch	Note
Analysis I		20022	10	□	Bestanden	1	0
Mathematik		20031	0	□	PV	1	0
Gdl. Algorithmen		20031	10	□	Bestanden	1	370
Analysis II		20041	10	☑	Bestanden	1	0
NP-Vollst Theo. & Anw I		20041	7.5	□	Bestanden	1	270
NP-Vollst Theo.&Anw. II		20041	7.5	□	Bestanden	1	270
M1: Mathematk1		20041	20	☑	Bestanden	1	100
Numerik I		20041	10	☑	Bestanden	1	0
Gdl. theo. Informatik		20041	10	□	Bestanden	1	330
M2: Mathematk		20051	20	☑	Bestanden	1	250

1 2 Page size: 10 **18** items in **2** pages

	Matrikel#	Exmatrikuliert	Bis	Noch Student	Fachwechsler	Bestanden	CP	8999	9000	Anzahl Prüfungen	Einschreibung
▸	1327805	☑	20041	□	□	☑	180			17	20022
▸	1474412	□	20111	□	□	☑	180			26	20022
▸	1513535	☑	20051	□	□	☑	180			21	20022
▸	1528701	☑	20061	□	□	☑	180			23	20022
▸	1532228	☑	20072	□	□	☑	180			22	20022
▸	1532260	☑	20051	□	□	☑	180			18	20022
▸	1532383	□	20081	□	□	☑	180			17	20022
▸	1536318	☑	20052	□	□	☑	180			24	20022
▸	1536509	☑	20051	□	□	☑	180			24	20022

1 2 3 4 5 6 7 8 9 10 ... Page size: 10 **424** items in **43** pages

Figure 4.3: Dashboard view for browsing and filtering the data

4.1.2.1 Visualizations

Park and Jo (2015) summarized different visualization techniques, which were used in examined learning analytics dashboards to visualize different types of information. These visualization techniques include bar graph, pie chart, table matrix, tag cloud, risk quadrant, scatter plot, win-lose chart, sociogram, timeline, line chart, signal lights, and wattle tree.

In this chapter, we summarize different types of visualization techniques which we propose to use in administrative educational dashboards and show examples based on the data set presented in Chapter 4.1.1.

Scatter plot diagrams

A scatter plot with connected points is suitable for depicting the number of dropouts in individual semesters. In Figure 4.4, both the dropouts per semester and the cumulative number of dropouts up to a particular semester are visualized. It becomes clear that most dropouts leave in the first semesters (as also observed by Fleischer et al. (2019)) and until the end of the fourth semester approx. 50% of all dropouts already left the course of studies. This shows the importance of early detection of dropouts in order to initiate interventions for at-risk students as early as possible. However, the other half of the dropouts leave the course of studies gradually.

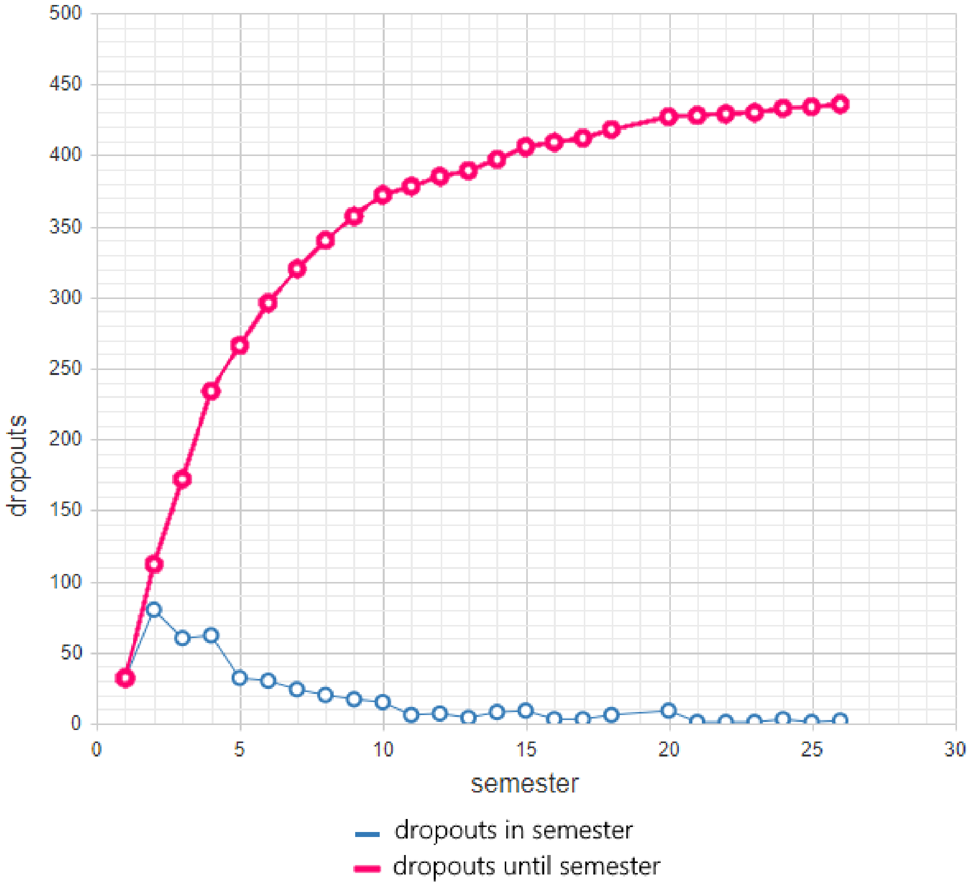


Figure 4.4: Dashboard view for the number of dropouts in individual semesters

Box plot diagrams

Box plots are for example suitable for a quick overview of the study duration of the different cohorts. Figure 4.5 shows the study duration of the graduates for different cohorts. The number 20062 on the y axis means that it is the cohort that started studying in the winter semester of 2006. Since the visualized data set comes from the year 2016, the study duration of the cohorts that started studying later is correspondingly shorter. This visualization shows that, on average, the students take longer to complete their studies than the 6 semesters specified as the standard period of study.

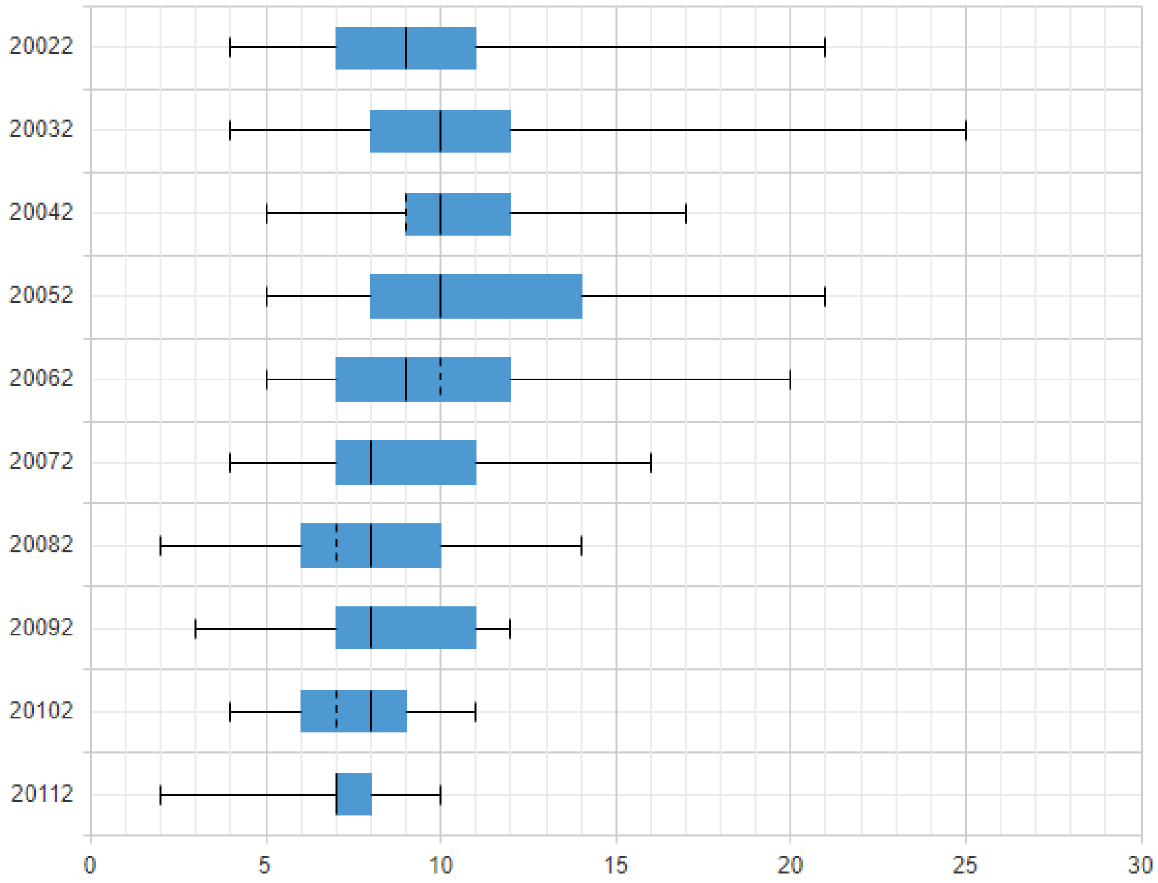
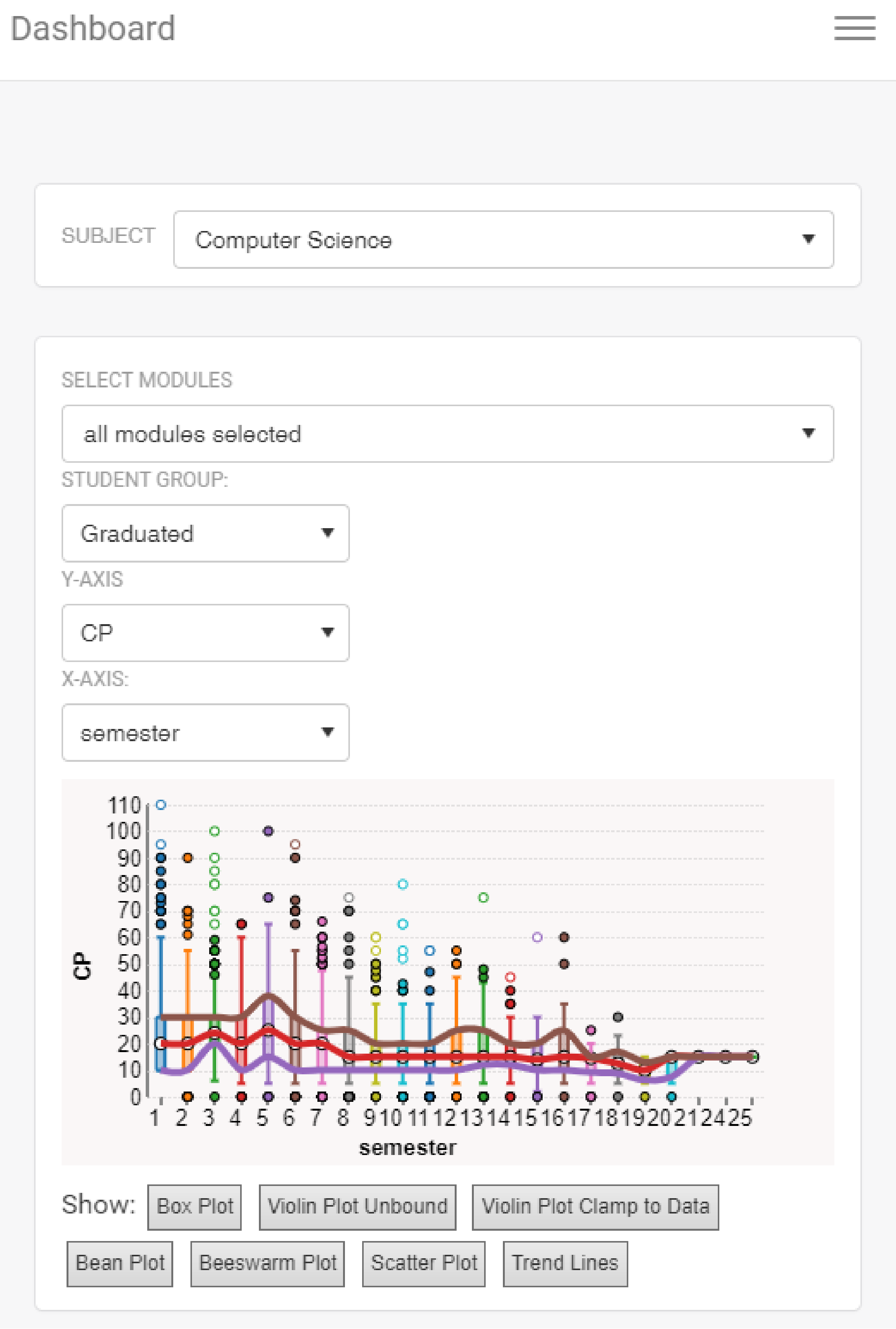


Figure 4.5: Dashboard view visualizing the number of semesters studied before graduation for different cohorts (students who started their studies in different semesters)

Figure 4.6 shows a dynamic visualization, which can be adjusted by filtering different parameters. In this example, it is visualized how many CP the graduates get in each semester. The red trend line shows that the graduates reach an average of 20 CP per semester, although approx. 30 CP per semester are specified. This explains the more extended study period of the graduates. The strong outliers in the CP can be explained by the recognized exams of some students.



Dashboard

SUBJECT
Computer Science

SELECT MODULES
all modules selected

STUDENT GROUP: Graduated

Y-AXIS
CP

X-AXIS: semester

Figure 4.6: Dashboard view for the amount of CP per semester of graduates

Sankey diagrams

There are not many visualization techniques in the literature that are tailored to specific characteristics of student data. In Askinadze et al. (2019b), we investigated which visualization techniques are suitable for viewing and comparing entire study processes of several cohorts at once. The visualization technique sankey diagram is well suited for this. In Figure 4.7, the study progress of dropouts who left the university by the end of the third semester and graduates is visualized in a Sankey diagram. The coding in the format *x_ yz* means: until the end of semester *x* this cohort has only passed the exams, which are indicated as yz in the legend. *k_ Dropout* represents the dropouts who are exmatriculated at the end of the *k*-th semester. For example, we can see that the group of students who passed both mathematics exams (Linear algebra and Calculus) at the end of the first semester almost exclusively consists of graduates. We can, therefore, learn from this presentation that passing the two mathematics exams in the first semester is a good indicator of a successful study (as shown in Chapter 3.4.3). On the other hand, dropouts who leave the course of study by the end of the third semester do not usually pass a single mathematics exam in the first semester. The proposed visualization with Sankey diagrams can be used for any number of semesters and exams. Figure 4.8 shows the Sankey diagram for the 10th semester and the three exams Calculus, Linear Algebra, and Programming.

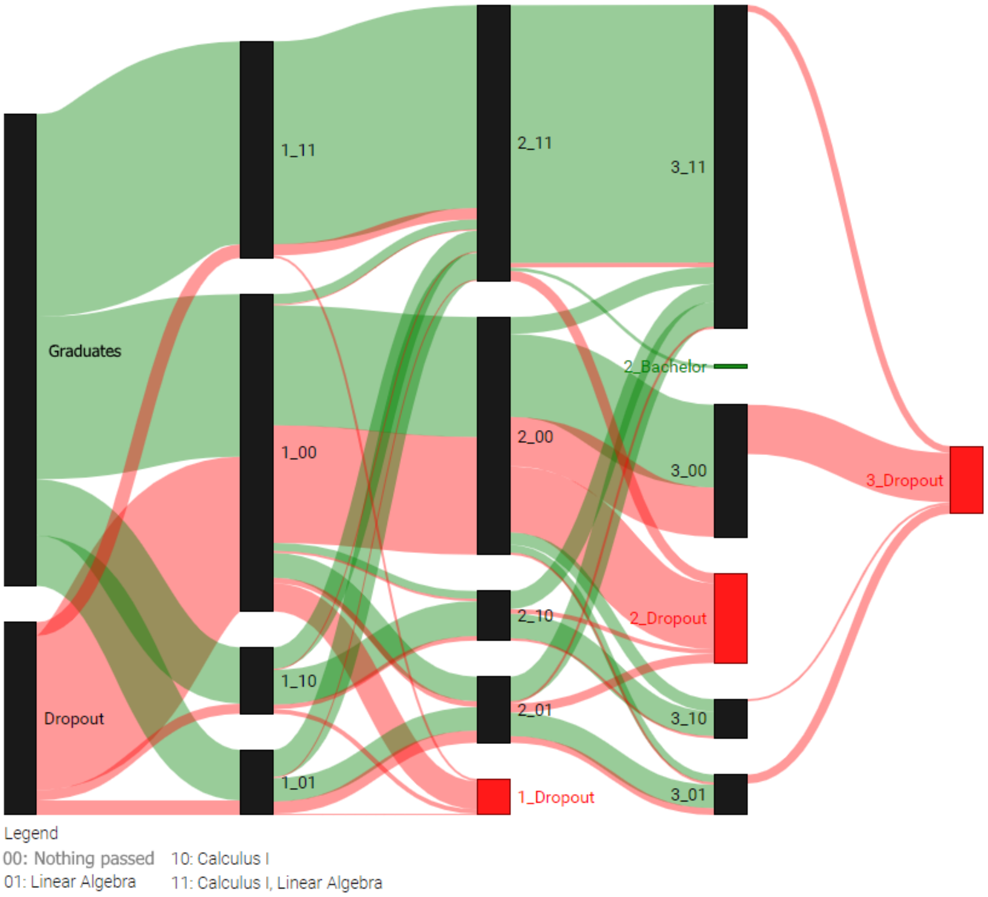


Figure 4.7: Dashboard view for Sankey diagrams (until 3rd semester)

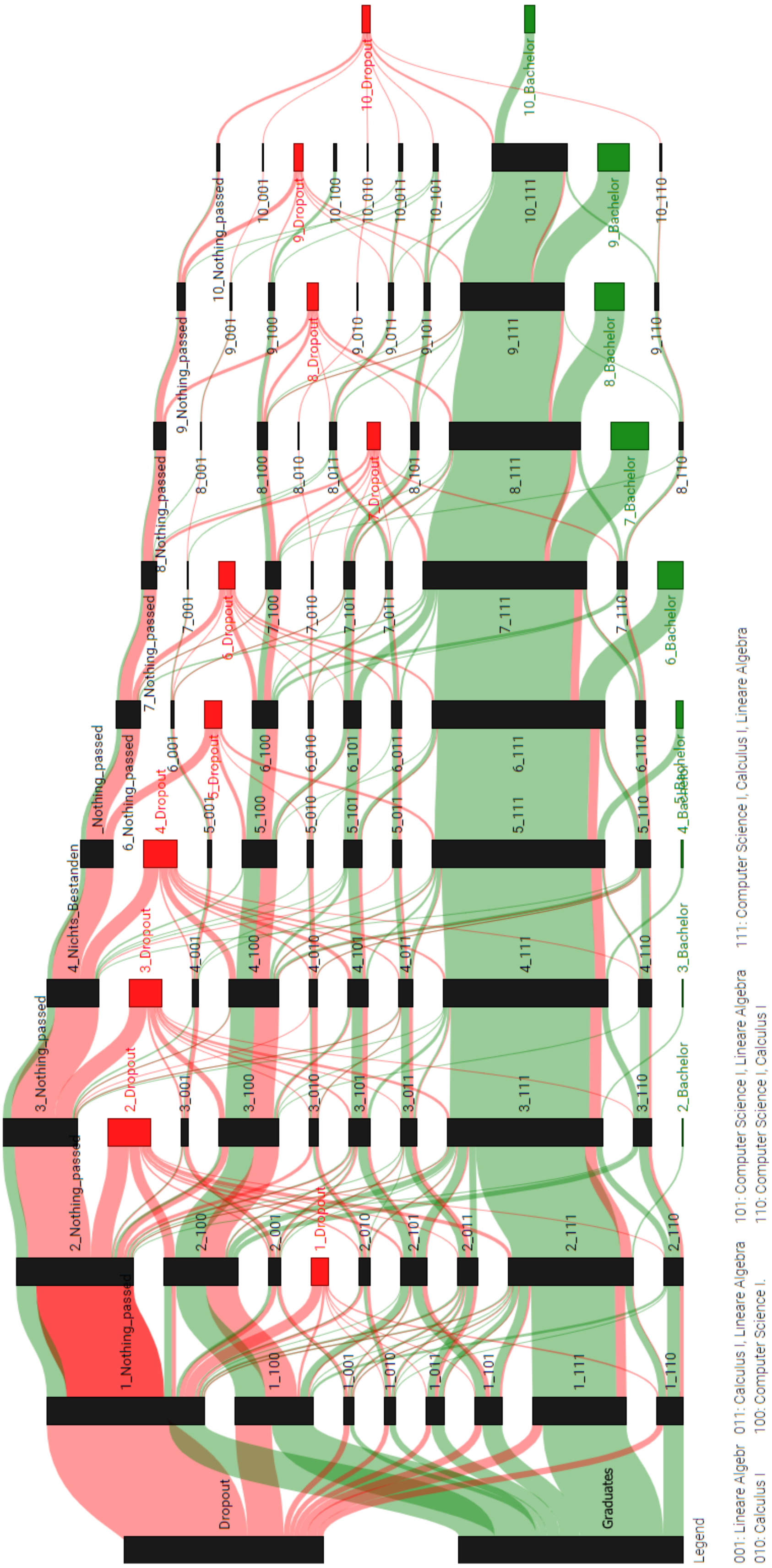


Figure 4.8: Dashboard view for Sankey diagrams (until 10th semester)

With the proposed visualization of the cohorts by Sankey diagrams, we can see the importance of the first semester exams for later academic success. The importance of the first semester for later academic success is also confirmed by Haarala-Muhonen et al. (2017) and our analysis in Chapter 3.4. Furthermore, we can find out which first semester exams are particularly important for later academic success.

Venn diagrams

In order to find out in which common exam combinations the students pass the exams up to a particular semester, Venn diagrams can be used (Askinadze et al., 2019b). Figure 4.9 shows, for example, for the students who have dropped out of university up to the 3rd semester, that in most cases, they do not pass both examinations. Linear Algebra, however, is passed more often than Calculus. The dropouts who are able to pass Calculus can usually also pass Linear Algebra. As we have seen, passing both mathematics exams is a good indicator of successful study. However, in this case, further exams should be included in the analysis to find out why these students drop out.

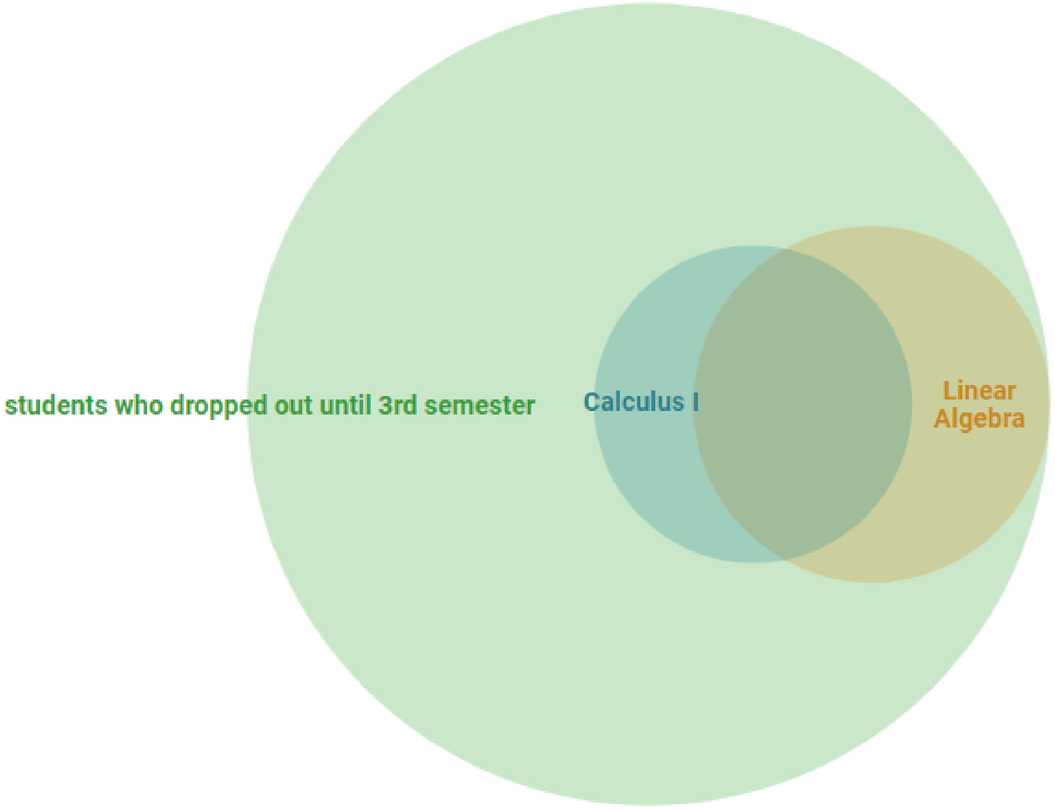


Figure 4.9: Venn diagram showing passed math exams of dropouts

4.1.2.2 Discussion

The visualizations presented can be an aid to decision-making, but do not make recommendations. They serve as an aid to various questions, such as (i) *how do different cohorts of students study?*, (ii) *what exams do different cohorts of students pass or fail?*, and (iii) *in which semesters are specific examinations passed?* The interpretation of the visualizations, therefore, requires human judgment, which must be made, for instance, by the university administration (Ionica, 2016; Büching et al., 2019).

4.2 Early Warning Systems

Early warning systems (EWS) are systems that should warn decision-makers of possible hazards so that they can prevent them. An EWS in combination with an administrative educational dashboard has the task of identifying students at risk of dropout or failing a course at an early stage in order to initiate measures that could prevent dropout (or failing a course) (Romero and Ventura, 2019; Grasso and Singh, 2011; Heppen and Therriault, 2008). These systems are the practical result of dropout prediction and performance prediction since they use the predictions to initiate interventions. Only a few studies so far have presented working EWS systems that had an impact on at-risk students (Arnold and Pistilli, 2012; Jayaprakash et al., 2014).

In Sclater et al. (2016), EWS were investigated, which are applied at universities. The authors mentioned that the use of LA is still in its infancy and EWS, which are used in practice, are mainly applied in North America, Australia, and Great Britain. In Table 4.2, we summarize some of the investigated EWS (which initiate interventions). The objectives of the individual systems can be roughly divided into two groups: *increasing retention* and *identifying students who need support*. The type of interventions is quite different: Posting traffic lights that indicate risk with color, sending e-mails informing about the danger of failing, talking to students about their situations, sending guidelines for improvement, forwarding students to online support and providing open educational resources, sending e-mails offering help, and making phone calls. Whether the interventions are really helpful has not yet been sufficiently researched. The results so far are: In the case of the University of New England, it is reported that the drop-out rate fell from 18% to 12%. At Marist College, it is reported that students to whom the interventions were applied received 6% better grades than students to whom no interventions were applied. In the case of the New York Institute of Technology (NYIT), it is reported that it was possible to predict the number of students at risk with a recall value of 74%. A similar recall value (75%) was achieved in the Marist College system.

	Purdue University	New York Institute of Technology	Marist College	Edith Cowan University	University of New England	Nottingham Trent University
Prediction of	student success at course level	dropout	students at risk of not completing the course	dropout	at-risk students	engagement
Goal	increasing retention and graduation rate	increasing retention in first year	provide students with feedback on their progress	identifying students who need support	identifying students who need support	enhance retention
Data source	student effort (VLE usage), prior academic history, student characteristics	admission application, registration test data, survey, financial data	demographic data, aptitude data, VLE usage	SIS, demographic data, progress information	students' emotion input, class attendance, study progress data, online portal usage	engagement data
Important features			previous grades, GPA, current academic standing			engagement is more important than background characteristics or entrance data
Information shown in student dashboard	traffic light				word cloud of aggregated students' comments	progress line indicating engagement, comparing students with rest of cohort
Information shown in staff dashboard	traffic light	table showing whether students return next year		report, probability score for each student	showing students in need of support; showing students' issues and concerns	same information as for students
Intervention	posting traffic light to students homepage or sending e-mail or arranging a meeting	possible conversation with each student about their situation	1) warning, message and guidance how to improve 2) directing to online support and providing open educational resources	personalised e-mails offering assistance, phone calls	automated mails followed by phone calls	mail if engagement stops for two weeks
Recall		74%	75% in 3 of 4 institutions			
Precision		55%				
Comments about success of the system	promising results, better grades		both strategies have same effect:both tratment groups have a 6% better grade compared to the group without intervention		dropout cut from 18% to 12%; positive student feedback	27% of students said they changed their bahvior based on data shown in dashboard

Table 4.2: Comparison of some EWS investigated in (Sclater et al., 2016), which are used in practice and initiate interventions.

4.3 Data Privacy

Since learning analytics and thus, the use of dashboards and early warning systems depend on legal requirements, we give an overview of the legal aspects in this chapter.

The use of learning analytics procedures raises ethical and data protection issues that, if ignored, can lead to negative consequences, as the following example shows: The nonprofit organization InBloom, founded in 2011, was sponsored by the Gates Foundation and Carnegie Corporation. The objective of the organization was to store, aggregate, and share data with trusted third parties. These third parties could then process the data, for example, to extract meaningful knowledge using LA methods. There were many negative voices from parents, lawyers, and teachers. For example, a parent representative said that InBloom was designed to "facilitate the sharing of children's personal and very sensitive information with data-mining vendors, with no attention paid to the need for parental notification or consent". The CEO of InBloom responded with "We do not actually control what data is uploaded" and "We open the vault for the district or state, they put the data in, and we lock it". After months of opposition from various parties against InBloom, the organization stopped the business (El-Khattabi, 2017; Herold, 2014).

The example above shows how important data protection laws are. In many countries, there are legal restrictions on the handling of private data. In the European Union, since 2018, the General Data Protection Regulation (GDPR) applies. However, compliance with GDPR is not only important for the EU. The website of the European Commission (European Commission, 2018) states that GDPR also applies outside the EU in two cases: "a company or entity which processes personal data as part of the activities of one of its branches established in the EU, regardless of where the data is processed; or (2) a company established outside the EU offering goods/services (paid or for free) or monitoring the behavior of individuals in the EU". That means any educational institution or private company that processes data from European students must take consideration of GDPR. Albrecht (2016) claimed that "GDPR now sets a standard that is to be taken as a clear statement by the biggest single market in the world. No data controller will be able to ignore this and other governments will be under pressure to raise their data protection standards in order to allow their economies access to the single digital market of the European Union". Buttarelli (2016) points out that "over half the countries in the world now have a data protection and/or privacy law, and most are strongly influenced by the European approach, a trend towards the 'global ubiquity' of data privacy." In Sabourin et al. (2015), the authors examined different legislations and concluded that the European Union had adopted comprehensive rules to protect the data of students, as these require a clear agreement from individuals to collect and process their personal information. The above sources show that methods are necessary which respect the regulations of GDPR while allowing the advantages of EDM and LA.

In literature, there are several discussions on how to handle data privacy in learning analytics. Often it is about allowing students to decide for themselves what is stored about them and that the data will not be passed on to third parties (El-Khattabi, 2017; Trainor, 2015). Few studies have presented frameworks and architectures that deal with current legal requirements (such as GDPR).

The Joint Information Systems Committee (JISC)[1] is a UK not-for-profit company that provides "trusted advice and practical assistance for universities, colleges, and learning providers" among other services. The JISC has published a guide and FAQs on how to use Learning Analytics and GDPR on its web pages (Sclater, 2018a; Sclater, 2018b). They explain that the consent of students does not make sense if no opt-out is possible. This applies in particular to data that is required for statistical reasons or to carry out education, such as date of birth, gender, modules, grades, previous qualifications, etc. Otherwise, the storage and processing of the data is regulated by Art. 6 of GDPR "Lawfulness of Processing" (European Union, 2016). Some important points that allow processing are:

- "the data subject has given consent to the processing of his or her personal data for one or more specific purposes;"
- "processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party [...]"
- "processing is necessary for compliance with a legal obligation to which the controller is subject;"
- "processing is necessary for the performance of a contract to which the data subject is party [...]"

In Sclater (2018b) it is pointed out, that the UK Information Commissioner's Office suggests "to avoid over-reliance on consent as a justification for data processing and that it's often better to use a different lawful basis." There are two exceptions that always require consent: (1) data from specific categories, such as ethnicity, and (2) analytics interventions with individual students. For interventions, for example, students must be asked by email if they want to. The following reasons are given by Sclater (2018a), why students should not always be asked to agree to the processing of their data in the context of learning analytics:

- "Using a justification such as "legitimate interests" for the processing of student data provides students with better safeguards than using consent. In this case the institution takes on the burden of ensuring that all such processing is done in ways, and subject to policies, that minimize the risk to individual students."
- "If consent is requested you freeze the activities to which it can apply: new types of analysis cannot be added if they were not envisaged at the time consent was obtained."
- "Enabling students to opt out of data collection may create 'holes' in the data set which reduce the effectiveness of learning analytics and disadvantage students overall"

[1] https://www.jisc.ac.uk/about/who-we-are-and-what-we-do

Cormack (2016) proposed to differentiate the purpose of the data processing in learning analytics to decide if consent is necessary or not. The author distinguishes between analysis and intervention:

- "discovery of significant patterns („analysis") treated as legitimate interest of the organization, which must include safeguards for individuals' interest and rights;"
- "the application of those patterns to meet the needs of particular individuals („intervention"), which requires their informed consent or, perhaps in future, an contractual agreement"

Cormack (2016) concluded that the "analysis of learning data is considered a legitimate interest of a university that must be conducted under appropriate safeguards" and therefore no explicit consent is needed, but "if analysis suggests an intervention that may affect students or staff, the consent of those individuals should be sought".

Cormack (2016) mentioned that anonymization or pseudonymization of data is an important topic. Only anonymized data should be used for the analysis. In cases where anonymization is not possible, pseudonymization methods should be used in which directly identifiable information is replaced or removed. In Khalil and Ebner (2016), the authors dealt with anonymization strategies for learning analytics. They listed *k-anonymity* (Sweeney, 2002) as one of the techniques. In k-anonymity, the data is changed so that the data of one student cannot be distinguished from the data of $k - 1$ other students. In Gursoy et al. (2017), different anonymization methods are evaluated. The authors stated in their discussion that in the majority of their experiments, anonymization methods such as k-anonymity offered results with higher utility and accuracy than those obtained using methods based on differential privacy (Dwork, 2008). The authors evaluated the methods on two data sets with real and synthetic data using different features to solve learning analytics tasks such as *grade point average* (GPA) prediction.

It has become clear that universities do not need the explicit consent of their students for data analysis if the reason for the analysis is well argued by the universities (Sclater, 2018b). Although this has the advantage of allowing significantly more data to be collected, which is important for learning analysis, it reduces the freedom of choice for students. JISC argues that this approach is better because, without explicit consent, universities are then obliged to develop all procedures in compliance with the law and better protect their students' data.

5 Conclusion

We have developed a dashboard which can integrate study progress data from different universities, regardless of whether the universities have hierarchical or non-hierarchical modules. To get an insight into how different student cohorts progress, we proposed different visualization approaches, such as Sankey, Venn, and Upset diagrams, that can be used to visualize the study progress data. With the help of the proposed visualizations, decision-makers are enabled to have a quick overview of study behavior. They can compare the study behavior of different cohorts to have a better basis for administrative decision making leading to better study experience. The visualizations can be helpful, but they do not give any recommendations, so the users have to interpret the images themselves. For example, we used a case study with a computer science course, and with the help of the Sankey diagrams we could see that many students find it challenging to pass Calculus and that students who pass Calculus in the first semester are most likely to graduate afterward.

In order to add an early warning system component to our administrative dashboard, we investigated how student dropout can be predicted from student progress data. We found that this works with high accuracy, recall, and precision (starting at 80% for all three evaluation measures) from the first semester onward. We introduced a new method that additionally includes a time component of the study progress data and thus provides slightly better prediction results. We also found that a small amount of data (only the information whether an exam was passed or not) is sufficient for prediction results that are similar to results presented in other recent papers.

For the research question on how to collect the data generated by students' interactions with digital learning elements and how to integrate the interaction data of different services of digital learning environments, we proposed an architecture based on the xAPI format that can integrate data from various heterogeneous data sources as a solution.

Based on several case studies, we have investigated how xAPI data can be used to predict student performance. On the one hand, we used interaction data from a learning management system and showed that the use of this data alone is not sufficient for good predictions, but in combination with different demographic data of the students, it provides comparatively good predictions. On the other hand, we used clickstream data from a digital eBook reader to predict the final grade of a test that was based on the eBook contents.

Since the application of learning analytics and educational data mining is not possible without considering data protection issues, we have investigated various recommendations for dealing with GDPR. In order for the use of LA and EDM to be possible, educational institutions must be open and transparent about what data is stored and for what purposes it is processed. If the purposes of processing are well-argued, student consent is not always necessary. However, if students have the choice which of their data may be tracked and stored, this leads to data sets with missing values. We have investigated the extent to which the missing values can be predicted from existing data in order to create better predictive models. We have found that even with comparatively high numbers of missing values, the evaluation results for student performance prediction decrease but remain acceptably good.

Outlook and future work

LA and EDM research in recent years has shown that the different types of student data can be used to support decision-makers through visualizations in administrative dashboards as well as to build early warning systems by predicting dropout and performance prediction. However, there is a lack of real systems in practice. In addition, there are currently only a few long-term studies on interventions that were initiated based on the predictions and to what extent these interventions finally led to changes in study behavior.

During the research for this thesis, we[1] developed a platform for the creation and management of digital and interactive learning elements (*Lernblitz*[2]), which can be used by educational institutions and is already used by thousands of students. The research done in this thesis on prediction on eBook behavioral usage data can be transferred to Lernblitz and used even more internally. Since Lernblitz, unlike an eBook, does not only display a sequence of content pages, but each page of a web-based training (WBT) can contain a quiz, interactive video, and many other interactive learning elements, data on the current state of knowledge of the students can be collected in addition to the pure clickstream behavior. As the research in this thesis has shown, data on past academic performance has the greatest influence on the prediction of future performance. Therefore, we assume that the combination of the learning elements clickstream behavior data, LMS behavior data, limited usage of demographic features and previous performance data (data from different intermediate tests in a WBT) will lead to an increase in predictive performance.

Our administrative educational dashboard for visualizing student progress data is one of the first of its kind and we will continue to develop it to offer it as a service to educational institutions. We have shown that the same data needed to visualize student progress is sufficient to make early predictions of student dropout in higher education. Therefore, we plan to extend the dashboard with an early warning component that will help institutional administrators to identify students who are at risk of dropout.

[1] In collaboration with *istis Informationssysteme* (`https://www.istis.de`)

[2] `https://www.lernblitz.de`

Abbreviations

ANN artificial neural network

ANOVA analysis of variance

AP average precision

API application programming interface

AUC area under the curve

CART classification and regression tree

CP credit points

CS computer science

DAG direct acyclic graph

DM data mining

DT decision tree

EDM educational data mining

EWS early warning system

FN false negative

FP false positive

FPR false positive rate

GDPR general data protection regulation

GFB global feature based

GPA grade point average

HStatG Higher Education Statistics Act

JISC joint information system committee

LA learning analytics

LAD learning analytics dashboard

LAK learning analytics & knowledge

LFB local feature based

LMS learning management system

MI mutual information

ML machine learning

MLP multi layer perceptron

PI performance indictor

QP quadratic programming

RBF radial basis function

REST representational state transfer

RF random forest

RMSE root mean square error

ROC receiver operator characteristic

RQ research question

SoLAR society for learning analytics research

SVM support vector machine

SVR support vector regression

TN true negative

TP true positive

VLE virtual learning environment

WBT web-based training

WSD weighted semester distance

xAPI experience API

References

Francesco Agrusti, Gianmarco Bonavolontà, and Mauro Mezzini (2019). "University Dropout Prediction through Educational Data Mining Techniques: A Systematic Review". In: *Journal of e-Learning and Knowledge Society* 15.3, pp. 161–182.

Gökhan Akçapınar, Mohammad Nehal Hasnine, Rwitajit Majumdar, Brendan Flanagan, and Hiroaki Ogata (2019). "Developing an early-warning system for spotting at-risk students by using eBook interaction logs". In: *Smart Learning Environments* 6.4, pp. 1–15.

Mayra Alban and David Mauricio (2019a). "Factors that Influence Undergraduate University Desertion According to Students Perspective". In: *International Journal of Engineering and Technology* 10.6, pp. 1585–1602.

Mayra Alban and David Mauricio (2019b). "Predicting University Dropout through Data Mining: A Systematic Literature". In: *Indian Journal of Science and Technology* 12.4, pp. 1–12.

Jan Philipp Albrecht (2016). "How the GDPR will change the world". In: *Eur. Data Prot. L. Rev.* 2, p. 287.

Hanan Aldowah, Hosam Al-Samarraie, and Wan Mohamad Fauzy (2019). "Educational Data Mining and Learning Analytics for 21st century higher education: A Review and Synthesis". In: *Telematics and Informatics*.

Elaf Abu Amrieh, Thair Hamtini, and Ibrahim Aljarah (2016). "Mining educational data to predict Student's academic performance using ensemble methods". In: *International Journal of Database Theory and Application* 9.8, pp. 119–136.

Kimberly E Arnold and Matthew D Pistilli (2012). "Course signals at Purdue: Using learning analytics to increase student success". In: *Proceedings of the 2nd international conference on learning analytics and knowledge*. ACM, pp. 267–270. DOI: 10.1145/2330601.2330666.

Alexander Askinadze (2016). "Anwendung der Regressions-SVM zur Vorhersage studentischer Leistungen". In: *Proceedings of the 28th GI-Workshop Grundlagen von Datenbanken, Nörten Hardenberg, Germany, May 24-27, 2016.* Pp. 15–20. URL: http://ceur-ws.org/Vol-1594/paper4.pdf.

Alexander Askinadze and Stefan Conrad (2017). "A Web Service Architecture for Tracking and Analyzing Data from Distributed E-Learning Environments". In: *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 208–213. DOI: 10.1109/WETICE.2017.52.

Alexander Askinadze and Stefan Conrad (2018a). "Development of an Educational Dashboard for the Integration of German State Universities' Data". In: *Proceedings of* *the 11th International Conference on Educational Data Mining, EDM 2018*. Buffalo NY, pp. 508–509.

Alexander Askinadze and Stefan Conrad (2018b). "Respecting Data Privacy in Educational Data Mining: An Approach to the Transparent Handling of Student Data and Dealing with the Resulting Missing Value Problem". In: *2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, pp. 160–164.

Alexander Askinadze and Stefan Conrad (2019). "Predicting Student Dropout in Higher Education Based on Previous Exam Results". In: *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019*. Montréal, Canada.

Alexander Askinadze, Matthias Liebeck, and Stefan Conrad (2018). "Predicting Student Test Performance based on Time Series Data of eBook Reader Behavior Using the Cluster-Distance Space Transformation". In: *Workshop Proceedings. 26th International Conference on Computers in Education (ICCE 2018)*. Manila, Philippines: Asia-Pacific Society for Computers in Education, pp. 430–439.

Alexander Askinadze, Matthias Liebeck, and Stefan Conrad (2019a). "BoB: A Bag of eBook Click Behavior Based Grade Prediction Approach". In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK'19)*, pp. 437–441.

Alexander Askinadze, Matthias Liebeck, and Stefan Conrad (2019b). "Using Venn, Sankey, and UpSet Diagrams to Visualize Students' Study Progress Based on Exam Combinations". In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK'19)*, pp. 759–763.

Lovenoor Aulck, Dev Nambi, Nishant Velagapudi, Joshua Blumenstock, and Jevin West (2019). "Mining University Registrar Records to Predict First-Year Undergraduate Attrition". In: pp. 9–18.

Diana Bajzek, William Brown, Marsha Lovett, and Gordon Rule (2007). "Inventing the digital dashboard for learning". In: *EdMedia+ Innovate Learning*. Association for the Advancement of Computing in Education (AACE), pp. 1084–1092.

Ryan SJD Baker and Kalina Yacef (2009). "The state of educational data mining in 2009: A review and future visions". In: *JEDM| Journal of Educational Data Mining* 1.1, pp. 3–17.

Jaroslav Bayer, Hana Bydzovská, Jan Géryk, Tomás Obsivac, and Lubomir Popelinsky (2012). "Predicting Drop-Out from Social Behaviour of Students." In: *International Educational Data Mining Society*.

Kristin P Bennett and Colin Campbell (2000). "Support vector machines: hype or hallelujah?" In: *Acm Sigkdd Explorations Newsletter* 2.2, pp. 1–13.

Johannes Berens, Kerstin Schneider, Simon Gortz, Simon Oster, and Julian Burghoff (2019). "Early detection of students at risk-predicting student dropouts using administrative student data from German universities and machine learning methods". In: *JEDM| Journal of Educational Data Mining* 11.3, pp. 1–41.

Daniel Bialecki (2013). *Booklet zur Studie Lernen mit Spaß*. URL: https://www-de.scoyo.com/dam/ratgeber-downloads/studie-lernen-mit-spass-booklet/booklet-lernen-mit-spass.pdf (visited on 11/27/2019).

Atena Bishka and Dylan Fedy (2018). *Are Textbooks Relevant in the Digital Learning Age?* URL: https://learningsolutionsmag.com/articles/are-textbooks-relevant-in-the-digital-learning-age (visited on 12/03/2019).

Christopher M Bishop (2006). *Pattern recognition and machine learning*. Springer.

María Blanca, Rafael Alarcón, Jaume Arnau, Roser Bono, and Rebecca Bendayan (2017). “Non-normal data: Is ANOVA still a valid option?” In: *Psicothema* 29.4, pp. 552–557.

Avrim L Blum and Pat Langley (1997). “Selection of relevant features and examples in machine learning”. In: *Artificial intelligence* 97.1-2, pp. 245–271.

L. Breiman, J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks.

Leo Breiman (2001). “Random forests”. In: *Machine learning* 45.1, pp. 5–32.

Leo Breiman et al. (1996). “Heuristics of instability and stabilization in model selection”. In: *The annals of statistics* 24.6, pp. 2350–2383.

Corinne Büching, Dana-Kristin Mah, Stephan Otto, Prisca Paulicke, and Ernst A Hartman (2019). “Learning Analytics an Hochschulen”. In: *Künstliche Intelligenz*. Springer, pp. 142–160.

Giovanni Buttarelli (2016). “The EU GDPR as a clarion call for a new global digital gold standard”. In: *International Data Privacy Law* 6.2, p. 77.

B. Chandra and Manish Gupta (2011). “An efficient statistical feature selection approach for classification of gene expression data”. In: *Journal of biomedical informatics* 44.4, pp. 529–535.

Chih-Chung Chang and Chih-Jen Lin (2011). “LIBSVM: A library for support vector machines”. In: *ACM Transactions on Intelligent Systems and Technology* 2 (3). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 27:1–27:27.

Maureen A Conard (2006). “Aptitude is not enough: How personality and behavior predict academic performance”. In: *Journal of Research in Personality* 40.3, pp. 339–346.

Albert T Corbett and John R Anderson (1994). “Knowledge tracing: Modeling the acquisition of procedural knowledge”. In: *User modeling and user-adapted interaction* 4.4, pp. 253–278.

Andrew Nicholas Cormack (2016). “A data protection framework for learning analytics”. In: *Journal of Learning Analytics* 3.1, pp. 91–106.

Corinna Cortes and Vladimir Vapnik (1995). “Support-vector networks”. In: *Machine learning* 20.3, pp. 273–297.

Paulo Cortez and Alice Silva (2008). “Using data mining to predict secondary school student performance”. In: *Proceedings of 5th Annual Future Business Technology Conference*. Ed. by A. Brito and J. Teixeira. EUROSIS, pp. 5–12.

George Cybenko (1989). “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Ben Daniel (2015). “Big Data and analytics in higher education: Opportunities and challenges”. In: *British journal of educational technology* 46.5, pp. 904–920.

Jesse Davis and Mark Goadrich (2006). “The relationship between Precision-Recall and ROC curves”. In: *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240.

Gerben W Dekker, Mykola Pechenizkiy, and Jan M Vleeshouwers (2009). “Predicting Students Drop Out: A Case Study.” In: *International Working Group on Educational Data Mining.*

Andreia Dionisio, Rui Menezes, and Diana A. Mendes (2004). “Mutual information: a measure of dependency for nonlinear time series”. In: *Physica A: Statistical Mechanics and its Applications* 344.1-2, pp. 326–329.

Pedro Domingos (2012). “A Few Useful Things to Know about Machine Learning”. In: *Commun. ACM* 55.10, pp. 78–87.

Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik (1997). “Support vector regression machines”. In: *Advances in neural information processing systems*, pp. 155–161.

Cynthia Dwork (2008). “Differential privacy: A survey of results”. In: *International Conference on Theory and Applications of Models of Computation*. Springer, pp. 1–19.

Tanya Elias (2011). “Learning analytics”. In: *Learning*, pp. 1–22.

European Commission (2018). URL: https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/application-regulation/who-does-data-protection-law-apply_en (visited on 12/01/2018).

European Union (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC (visited on 05/05/2020).

Stephen Few (2004). “Dashboard confusion”. In: *Intelligent Enterprise* 7.4, pp. 14–15.

C.T. Fitz-Gibbon (1990). *Performance Indicators*. BERA Dialogues Series. Multilingual Matters. ISBN: 9781853590924. URL: https://books.google.de/books?id=uxKOMUHeiI4C.

Brendan Flanagan, Weiqin Chen, and Hiroaki Ogata (2018). “Joint Activity on Learner Performance Prediction using the BookRoll Dataset”. In: *International Conference on Computers in Education (ICCE2018)*, pp. 480–485.

Brendan Flanagan and Hiroaki Ogata (2017). “Integration of learning analytics research and production systems while protecting privacy”. In: *The 25th International Conference on Computers in Education, Christchurch, New Zealand*, pp. 333–338.

Brendan Flanagan, Atsushi Shimada, Stephen Yang, Bae-Ling Chen, Yang-Chia Shih, and Hiroaki Ogata (2019). “Predicting Performance Based on the Analysis of Reading Behavior: A Data Challenge”. In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK’19)*, pp. 759–763.

Jens Fleischer, Detlev Leutner, Matthias Brand, Hans Fischer, Martin Lang, Philipp Schmiemann, and Elke Sumfleth (2019). “Vorhersage des Studienabbruchs in naturwissenschaftlich-technischen Studiengängen”. In: *Zeitschrift für Erziehungswissenschaft*, pp. 1–21.

Aurélien Géron (2018). *Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme*. O’Reilly.

Veronica F. Grasso and Ashbindu Singh (2011). “Early warning systems: State-of-art analysis and future directions”. In: *Draft report, UNEP* 1.

Mehmet Emre Gursoy, Ali Inan, Mehmet Ercan Nergiz, and Yucel Saygin (2017). “Privacy-preserving learning analytics: challenges and techniques”. In: *IEEE Transactions on Learning technologies* 10.1, pp. 68–81.

Isabelle Guyon and André Elisseeff (2003). “An introduction to variable and feature selection”. In: *Journal of machine learning research* 3.Mar, pp. 1157–1182.

Anne Haarala-Muhonen, Mirja Ruohoniemi, Anna Parpala, Erkki Komulainen, and Sari Lindblom-Ylänne (2017). “How do the different study profiles of first-year students predict their study success, study progress and the completion of degrees?” In: *Higher Education* 74.6, pp. 949–962.

Jun Han and Claudio Moraga (1995). “The influence of the sigmoid function parameters on the speed of backpropagation learning”. In: *International Workshop on Artificial Neural Networks*. Springer, pp. 195–201.

Karin Hartl (2019). “The Application Potential of Data Mining in Higher Education Management: A Case Study Based on German Universities”. PhD thesis. Karlsruher Institut für Technologie (KIT). 177 pp. DOI: 10.5445/IR/1000096613.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Jessica B. Heppen and Susan Bowles Therriault (2008). “Developing Early Warning Systems to Identify Potential High School Dropouts. Issue Brief.” In: *National High School Center*.

Benjamin Herold (2014). *inBloom to Shut Down Amid Growing Data-Privacy Concerns*. URL: http://blogs.edweek.org/edweek/DigitalEducation/2014/04/inbloom_to_shut_down_amid_growing_data_privacy_concerns.html (visited on 11/09/2018).

Ulrich Heublein, Christopher Hutzsch, Jochen Schreiber, Dieter Sommer, and Georg Besuch (2007). “Ursachen des Studienabbruchs in Bachelor-und in herkömmlichen Studiengängen”. In: *Ergebnisse einer bundesweiten Befragung von Exmatrikulierten des Studienjahres* 8.2.

Pascal Hirmer, Tim Waizenegger, Ghareeb Falazi, Majd Abdo, Yuliya Volga, Alexander Askinadze, Matthias Liebeck, Stefan Conrad, Tobias Hildebrandt, Conrad Indiono, Stefanie Rinderle-Ma, Martin Grimmer, Matthias Kricke, and Eric Peukert (2017). “The First Data Science Challenge at BTW 2017”. In: *Datenbank-Spektrum* 17.3, pp. 207–222.

Sachio Hirokawa and Chengjiu Yin (2019). “Feature Engineering for Learning Log Analysis”. In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK’19)*, pp. 484–491.

Tin Kam Ho (1995). “Random decision forests”. In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, pp. 278–282.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola (2008). “Kernel methods in machine learning”. In: *The annals of statistics*, pp. 1171–1220.

Elke Hörnstein, Horst Kreth, Christian Blank, and Carolin Stellmacher (2016). *Studiengang-Monitoring: Studienverlaufsanalysen auf Basis von ECTS-Punkten*. Shaker.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin (2003). *A Practical Guide to Support Vector Classification*. Tech. rep. Department of Computer Science, National Taiwan University. URL: http://www.csie.ntu.edu.tw/~cjlin/papers.html.

Chih-Wei Hsu and Chih-Jen Lin (2002). “A comparison of methods for multiclass support vector machines”. In: *IEEE transactions on Neural Networks* 13.2, pp. 415–425.

Ya-Han Hu, Chia-Lun Lo, and Sheng-Pao Shih (2014). "Developing early warning systems to predict students' online learning performance". In: *Computers in Human Behavior* 36, pp. 469–478.

Lavina Ionica (2016). *Learning analytics in der Hochschullehre.* URL: `https://www.hochschulforumdigitalisierung.de/de/blog/learning-analytics-hochschullehre` (visited on 12/03/2019).

Sandeep M. Jayaprakash, Erik W. Moody, Eitel JM Lauría, James R. Regan, and Joshua D. Baron (2014). "Early alert of academically at-risk students: An open source analytics initiative". In: *Journal of Learning Analytics* 1.1, pp. 6–47.

Makhlouf Jihed and Tsunenori Mine (2019). "Investigating Reading Behaviors within Student Reading Sessions to Predict their Performance". In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK'19)*, pp. 455–464.

Mohammad Khalil and Martin Ebner (2016). "De-identification in learning analytics". In: *Journal of Learning Analytics* 3.1, pp. 129–138.

Meriem El-Khattabi (2017). "Mining for Success: Have Student Data Privacy and Educational Data Mining Created a Legislative War Zone". In: *Journal of Law, Technology & Policy*, pp. 511–538.

Hae-Young Kim (2014). "Analysis of variance (ANOVA) comparing means of more than two groups". In: *Restorative dentistry & endodontics* 39.1, pp. 74–77.

Andy Kirk (2012). *Data Visualization: a successful design process.* Packt Publishing Ltd.

Joris Klerkx, Katrien Verbert, and Erik Duval (2017). "Learning Analytics Dashboards". In: *Handbook of Learning Analytics.* SOLAR, Society for Learning Analytics and Research, pp. 143–150.

S. Klöpping, M. Scherfer, S. Gokus, A. Dachsberger, A. Krieg, A. Wolter, R. Bruder, W. Ressel, and E. Umbach (2017). "Studienabbruch in den Ingenieurwissenschaften". In: *Empirische Analyse und best practices zum Studienerfolg.*

Ron Kohavi and George H. John (1997). "Wrappers for feature subset selection". In: *Artificial intelligence* 97.1-2, pp. 273–324.

Meera Komarraju and Steven J Karau (2005). "The relationship between the big five personality traits and academic motivation". In: *Personality and individual differences* 39.3, pp. 557–567.

Meera Komarraju, Steven J. Karau, and Ronald R. Schmeck (2009). "Role of the Big Five personality traits in predicting college students' academic motivation and achievement". In: *Learning and individual differences* 19.1, pp. 47–52.

Meera Komarraju, Steven J. Karau, Ronald R. Schmeck, and Alen Avdic (2011). "The Big Five personality traits, learning styles, and academic achievement". In: *Personality and individual differences* 51.4, pp. 472–477.

Michael Søgaard Larsen, Kasper Pihl Kornbeck, Rune Müller Kristensen, Malene Rode Larsen, and Hanna Bjørnøy Sommersel (2012). "Dropout Phenomena at Universities: What is Dropout? Why does". In: *Education* 45, pp. 1111–1120.

Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann (2014). "Industry 4.0". In: *Business & information systems engineering* 6.4, pp. 239–242.

Yann LeCun, Leon Bottou, Genevieve B Orr, and Klaus-Robert Müller (1998). "Efficient BackProp". In: *Neural Networks: Tricks of the Trade.* Springer, pp. 9–50.

Matthias Liebeck, Pashutan Modaresi, Alexander Askinadze, and Stefan Conrad (2016). "Pisco: A Computational Approach to Predict Personality Types from Java Source Code". In: *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation*, pp. 43–47.

Owen H.T. Lu, Anna Y.Q. Huang, and Stephen. J.H Yang (2018). "Benchmarking and Tuning Regression Algorithms on Predicting Students' Academic Performance". In: *Workshop Proceedings. 26th International Conference on Computers in Education (ICCE 2018)*. Manila, Philippines: Asia-Pacific Society for Computers in Education, pp. 477–486.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Rubén Manrique, Bernardo Pereira Nunes, Olga Mariño, Marco Antonio Casanova, and Terhi Nurmikko-Fuller (2019). "An Analysis of Student Representation, Representative Features and Classification Algorithms to Predict Degree Dropout". In: *Proceedings of the 9th International Conference on Learning Analytics & Knowledge, LAK 2019, Tempe, AZ, USA, March 4-8, 2019*, pp. 401–410. DOI: `10.1145/3303772.3303800`.

Warren S McCulloch and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Alan G. Mencher (1971). "On the Social Deployment of Science". In: *Bulletin of the Atomic Scientists* 27.10, pp. 34–38. DOI: `10.1080/00963402.1971.11455425`.

Martijn Millecamp, Tom Broos, Tinne De Laet, and Katrien Verbert (2019). "DIY: learning analytics dashboards". In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK'19)*. Solar, pp. 947–954.

Tom M. Mitchell (1997). *Machine Learning*. New York: McGraw-Hill. ISBN: 978-0-07-042807-2.

Burkhart Müller (1985). "INSTITUTIONAL ADMINISTRATORS". In: *Higher Education in Europe* 10.3, pp. 19–24. DOI: `10.1080/0379772850100304`.

Joachim Müller, Anita Stender, Jens Fleischer, Andreas Borowski, Elmar Dammann, Martin Lang, and Hans E. Fischer (2018). "Mathematisches Wissen von Studienanfängern und Studienerfolg". In: *Zeitschrift für Didaktik der Naturwissenschaften* 24.1, pp. 183–199. ISSN: 2197-988X. DOI: `10.1007/s40573-018-0082-y`.

Kevin P Murphy (2012). *Machine learning: a probabilistic perspective*. MIT press.

Ashutosh Nandeshwar, Tim Menzies, and Adam Nelson (2011). "Learning patterns of university student retention". In: *Expert Systems with Applications* 38.12, pp. 14984–14996.

National Association of College Stores (2018). *Highlights from Student Watch Attitudes & Behaviors toward Course Materials 2017-18 Report*. URL: `http://www.nacs.org/research/studentwatchfindings.aspx` (visited on 05/05/2020).

Kelvin H. R. Ng, Sivanagaraja Tatinati, and Andy W. H. Khong (2019). "Characterization of Fuzziness for Grade Prediction using Deep Neural Networks". In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK'19)*, pp. 448–454.

José María Ortiz-Lozano, Antonio Rua-Vieites, Paloma Bilbao-Calabuig, and Martí Casadesús-Fa (2018). "University student retention: Best time and data to identify undergraduate students at risk of dropout". In: *Innovations in Education and Teaching International*, pp. 1–12.

Rutvija Pandya and Jayati Pandya (2015). “C5.0 algorithm to improved decision tree with feature selection and reduced error pruning”. In: *International Journal of Computer Applications* 117.16, pp. 18–21.

Su-lin Pang and Ji-zhang Gong (2009). “C5.0 classification algorithm and application on individual credit evaluation of banks”. In: *Systems Engineering-Theory & Practice* 29.12, pp. 94–104.

Yeonjeong Park and I-H Jo (2015). “Development of the learning analytics dashboard to support students’ learning performance”. In: *Journal of Universal Computer Science* 21.1, p. 110.

Christy Pettey (2011). *Gartner Says Worldwide Enterprise IT Spending to Reach $2.7 Trillion in 2012*. URL: `https://web.archive.org/web/20170612085034/http://www.gartner.com/newsroom/id/1824919` (visited on 04/29/2020).

John C. Platt (1998). *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. Tech. rep. MSR-TR-98-14. URL: `https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/`.

John C. Platt, Nello Cristianini, and John Shawe-Taylor (2000). “Large margin DAGs for multiclass classification”. In: *Advances in neural information processing systems*, pp. 547–553.

J. R. Quinlan (1986). “Induction of Decision Trees”. In: *Machine learning* 1.1, pp. 81–106.

J. R. Quinlan (1987). “Generating Production Rules from Decision Trees”. In: *Proceedings of the 10th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI’87. Milan, Italy: Morgan Kaufmann Publishers Inc., pp. 304–307.

J. R. Quinlan (1993). *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1-55860-238-0. URL: `http://portal.acm.org/citation.cfm?id=152181`.

J. R. Quinlan (2018). *C5*. URL: `https://rulequest.com/`.

Qasem A Al-Radaideh, Emad M Al-Shawakfa, and Mustafa I Al-Najjar (2006). “Mining student data using decision trees”. In: *International Arab Conference on Information Technology (ACIT’2006), Yarmouk University, Jordan*.

Francisco Rangel, Fabio González, Felipe Restrepo, Manuel Montes, and Paolo Rosso (2016). “Pan@ fire: overview of the pr-soco track on personality recognition in source code”. In: *Forum for Information Retrieval Evaluation*. Springer, pp. 1–19.

Rat für kulturelle Bildung (2019). *JUGEND / YOUTUBE / KULTURELLE BILDUNG. HORIZONT 2019*. Studie. Rat für kulturelle Bildung. URL: `https://www.rat-kulturellebildung.de/fileadmin/user_upload/pdf/Studie_YouTube_Webversion_final.pdf` (visited on 05/05/2020).

Russell D. Reed and Robert J. Marks (1999). *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press.

Cristobal Romero and Sebastian Ventura (2007). “Educational data mining: A survey from 1995 to 2005”. In: *Expert systems with applications* 33.1, pp. 135–146.

Cristobal Romero and Sebastian Ventura (2010). “Educational data mining: a review of the state of the art”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.6, pp. 601–618.

Cristobal Romero and Sebastian Ventura (2013). “Data mining in education”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3.1, pp. 12–27.

Cristobal Romero and Sebastian Ventura (2019). "Guest Editorial: Special Issue on Early Prediction and Supporting of Learning Performance". In: *IEEE Transactions on Learning Technologies* 12.2, pp. 145–147. ISSN: 1939-1382. DOI: 10.1109/TLT.2019.2908106.

Cristobal Romero, Sebastian Ventura, Pedro Espejo, and Cesar Martínez (2008). "Data Mining Algorithms to Classify Students." In: *Proceedings of the 1st International Conference on Educational Data Mining, EDM 2008*. Montréal, Canada, pp. 8–17.

Frank Rosenblatt (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6, p. 386.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536.

Graeme D. Ruxton (2006). "The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test". In: *Behavioral Ecology* 17.4, pp. 688–690. ISSN: 1045-2249. DOI: 10.1093/beheco/ark016. eprint: http://oup.prod.sis.lan/beheco/article-pdf/17/4/688/17275561/ark016.pdf.

Amjed Abu Saa, Mostafa Al-Emran, and Khaled Shaalan (2019). "Factors Affecting Students' Performance in Higher Education: A Systematic Review of Predictive Data Mining Techniques". In: *Technology, Knowledge and Learning*, pp. 1–32.

Jennifer Sabourin, Lucy Kosturko, Clare Fitzgerald, and Scott W. McQuiggan (2015). "Student Privacy and Educational Data Mining: Perspectives from Industry". In: *Proceedings of the 8th International Conference on Educational Data Mining*, pp. 164–170.

Mark Schneider (2010). "Finishing the First Lap: The Cost of First Year Student Attrition in America's Four Year Colleges and Universities." In: *American Institutes for Research.*

Beat A. Schwendimann, Maria Jesus Rodriguez-Triana, Andrii Vozniuk, Luis P. Prieto, Mina Shirvani Boroujeni, Adrian Holzer, Denis Gillet, and Pierre Dillenbourg (2016). "Understanding Learning at a Glance: An Overview of Learning Dashboard Studies". In: *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. LAK '16. Edinburgh, United Kingdom: ACM, pp. 532–533. ISBN: 978-1-4503-4190-5. DOI: 10.1145/2883851.2883930.

Niall Sclater (2018a). *GDPR and Learning Analytics – Frequently Asked Questions.* URL: https://analytics.jiscinvolve.org/wp/2018/06/01/gdpr-and-learning-analytics-frequently-asked-questions/ (visited on 12/01/2018).

Niall Sclater (2018b). *Learning analytics and GDPR: what you need to know.* URL: https://www.jisc.ac.uk/blog/learning-analytics-and-gdpr-what-you-need-to-know-20-sep-2018 (visited on 12/01/2018).

Niall Sclater, Alice Peasgood, and Joel Mullan (2016). "Learning analytics in higher education". In: *London: Jisc.* 8, p. 2017.

Amirah Mohamed Shahiri, Wahidah Husain, and Nur'aini Abdul Rashid (2015). "A review on predicting student's performance using data mining techniques". In: *Procedia Computer Science* 72, pp. 414–422.

George Siemens and Ryan SJD Baker (2012). "Learning Analytics and Educational Data Mining: Towards Communication and Collaboration". In: *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*. LAK '12. Vancouver, British Columbia, Canada: ACM, pp. 252–254. ISBN: 978-1-4503-1111-3. DOI: 10.1145/2330601.2330661.

Reginald Smith (2015). "A mutual information approach to calculating nonlinearity". In: *Stat* 4.1, pp. 291–303.

Alex J. Smola and Bernhard Schölkopf (2004). "A tutorial on support vector regression". In: *Statistics and computing* 14.3, pp. 199–222.

William G. Spady (1970). "Dropouts from higher education: An interdisciplinary review and synthesis". In: *Interchange* 1.1, pp. 64–85. ISSN: 1573-1790. DOI: 10.1007/BF02214313.

Latanya Sweeney (2002). "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05, pp. 557–570.

Gineke A. Ten Holt, Marcel JT Reinders, and EA Hendriks (2007). "Multi-dimensional dynamic time warping for gesture recognition". In: *Thirteenth annual conference of the Advanced School for Computing and Imaging*. Vol. 300, p. 1.

Vincent Tinto (1982). "Limits of theory and practice in student attrition". In: *The journal of higher education* 53.6, pp. 687–700.

Sonja Trainor (2015). "Student data privacy is cloudy today, clearer tomorrow". In: *Phi Delta Kappan* 96.5, pp. 13–18.

Sabrina Trapmann, Benedikt Hell, Sonja Weigand, and Heinz Schuler (2007). "Die Validität von Schulnoten zur Vorhersage des Studienerfolgs-eine Metaanalyse". In: *Zeitschrift für pädagogische Psychologie* 21.1, pp. 11–27.

Stuart A Tross, Jeffrey P Harper, Lewis W Osher, and Linda M Kneidinger (2000). "Not just the usual cast of characteristics: Using personality to predict college performance and retention." In: *Journal of College Student Development*.

Jake VanderPlas (2016). *Python data science handbook: essential tools for working with data*. " O'Reilly Media, Inc."

Katrien Verbert, Sten Govaerts, Erik Duval, Jose Luis Santos, Frans Assche, Gonzalo Parra, and Joris Klerkx (2014). "Learning dashboards: an overview and future research opportunities". In: *Personal and Ubiquitous Computing* 18.6, pp. 1499–1514.

A.R. Webb and K.D. Copsey (2011). *Statistical Pattern Recognition*. Wiley. ISBN: 9781119952961. URL: https://books.google.es/books?id=WpV9Xt-h3OOC.

List of Figures

[3] The visualization is motivated by http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

List of Tables

A

Publications

A.1 A Web Service Architecture for Tracking and Analyzing Data from Distributed E-Learning Environments

Alexander Askinadze and Stefan Conrad (2017). “A Web Service Architecture for Tracking and Analyzing Data from Distributed E-Learning Environments”. In: *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 208–213. DOI: 10.1109/WETICE.2017.52.

The research and the preparation of the manuscript was done entirely by Alexander Askinadze under the supervision of Stefan Conrad.

Status: Published.

A.2 Development of an Educational Dashboard for the Integration of German State Universities' Data

Alexander Askinadze and Stefan Conrad (2018a). "Development of an Educational Dashboard for the Integration of German State Universities' Data". In: *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018*. Buffalo NY, pp. 508–509.

The research and the preparation of the manuscript was done entirely by Alexander Askinadze under the supervision of Stefan Conrad.

Status: Published.

A.3 Using Venn, Sankey, and UpSet Diagrams to Visualize Students' Study Progress Based on Exam Combinations

Alexander Askinadze, Matthias Liebeck, and Stefan Conrad (2019b). "Using Venn, Sankey, and UpSet Diagrams to Visualize Students' Study Progress Based on Exam Combinations". In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK'19)*, pp. 759–763.

Contributions: The code and research was done entirely by Alexander Askinadze. The manuscript was prepared jointly by Alexander Askinadze and Matthias Liebeck under the supervision of Stefan Conrad.

Status: Published.

A.4 Predicting Student Test Performance based on Time Series Data of eBook Reader Behavior Using the Cluster-Distance Space Transformation

Alexander Askinadze, Matthias Liebeck, and Stefan Conrad (2018). “Predicting Student Test Performance based on Time Series Data of eBook Reader Behavior Using the Cluster-Distance Space Transformation”. In: *Workshop Proceedings. 26th International Conference on Computers in Education (ICCE 2018)*. Manila, Philippines: Asia-Pacific Society for Computers in Education, pp. 430–439.

Contributions: The research was conducted jointly by Alexander Askinadze and Matthias Liebeck. The manuscript was prepared jointly by Alexander Askinadze and Matthias Liebeck under the supervision of Stefan Conrad.

Status: Published.

A.5 BoB: A Bag of eBook Click Behavior Based Grade Prediction Approach

Alexander Askinadze, Matthias Liebeck, and Stefan Conrad (2019a). "BoB: A Bag of eBook Click Behavior Based Grade Prediction Approach". In: *Companion Proceeding of the 9th International Conference on Learning Analytics & Knowledge (LAK'19)*, pp. 437–441.

Contributions: The research was conducted jointly by Alexander Askinadze and Matthias Liebeck. The manuscript was prepared jointly by Alexander Askinadze and Matthias Liebeck under the supervision of Stefan Conrad.

Status: Published.

A.6 Predicting Student Dropout In Higher Education Based on Previous Exam Results

Alexander Askinadze and Stefan Conrad (2019). “Predicting Student Dropout in Higher Education Based on Previous Exam Results”. In: *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019.* Montréal, Canada.

Contributions: The research and the preparation of the manuscript was done entirely by Alexander Askinadze under the supervision of Stefan Conrad.

Status: Published.

A.7 Respecting Data Privacy in Educational Data Mining: An Approach to the Transparent Handling of Student Data and Dealing with the Resulting Missing Value Problem

Alexander Askinadze and Stefan Conrad (2018b). “Respecting Data Privacy in Educational Data Mining: An Approach to the Transparent Handling of Student Data and Dealing with the Resulting Missing Value Problem”. In: *2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, pp. 160–164.

Contributions: The research and the preparation of the manuscript was done entirely by Alexander Askinadze under the supervision of Stefan Conrad

Status: Published.

A.8 The First Data Science Challenge at BTW 2017

Pascal Hirmer, Tim Waizenegger, Ghareeb Falazi, Majd Abdo, Yuliya Volga, Alexander Askinadze, Matthias Liebeck, Stefan Conrad, Tobias Hildebrandt, Conrad Indiono, Stefanie Rinderle-Ma, Martin Grimmer, Matthias Kricke, and Eric Peukert (2017). “The First Data Science Challenge at BTW 2017”. In: *Datenbank-Spektrum* 17.3, pp. 207–222.

Contributions: Alexander Askinadze and Matthias Liebeck participated in the BTW 2017 Data Science Challenge and achieved 2$^{\text{nd}}$ place which was awarded with 300 € prize money. The third section of the manuscript was prepared jointly by Alexander Askinadze and Matthias Liebeck under the supervision of Stefan Conrad.

Status: Published.

A.9 Pisco: A Computational Approach to Predict Personality Types from Java Source Code

Matthias Liebeck, Pashutan Modaresi, Alexander Askinadze, and Stefan Conrad (2016). "Pisco: A Computational Approach to Predict Personality Types from Java Source Code". In: *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation*, pp. 43–47.

Contributions: Matthias Liebeck and Pashutan Modaresi applied pair programming to create the code. Alexander Askinadze participated in the feature engineering. The manuscript was prepared jointly by Matthias Liebeck, Pashutan Modaresi, and Alexander Askinadze under the supervision of Stefan Conrad.

Status: Published.