Scalable Interventions in Higher Education: An AI-Supported Smartphone App to ASSIST Self-Reflection

Johannes Berens[a], Thomas Gößl[b], Leandro Henao[b], Luis Rumert[b;*], Moritz Salamon[b] and Kerstin Schneider[c]

[a]RFH Köln and University of Wuppertal
[b]University of Wuppertal
[c]University of Wuppertal and CESifo

Abstract. ASSIST is an intelligent feedback portal for smartphones that helps students monitor their academic progress and predict their likelihood of success. In addition to providing real-time data on their performance (number of completed credit points and average grades) and comparison to their peers, it allows students to reflect on their current situation and explore individualized scenarios to make them aware of the efforts required for successful completion. By leveraging standardized student data, including demographic and academic information, ASSIST can be integrated in any higher education institution in Germany. In this contribution, we highlight the relevance of AI-supported feedback tools to foster self-reflection among the students and provide details on our data and the neural network used to compute success probabilities, as well as the technical characteristics of this application.

1 Introduction

While the number of new enrollments per semester has increased in Germany from 300,000 in 2007 to about 400,000 in 2022, the concern regarding high dropout rates and slow completion remains. As of 2020, 35% of German students in Bachelor programs leave the university without a degree, and this share is even more significant for students who obtained their entry qualification outside of Germany [8]. In the face of this challenge to the efficient allocation of resources and the supply of qualified workers, the focus has been placed on promoting study success and avoiding failure.

In this regard, one of the major contributions from the field of educational data mining and learning analytics has been the implementation of dropout prediction systems (see, e.g., [1, 2, 4, 11]). However, even though these tools exhibit high levels of prediction accuracy and thus open the door to targeted interventions for at-risk students, more is needed. Indeed, many times such treatments barely lead to any behavioral adjustment among students [13] or only help to improve academic outcomes among already successful students [3]; in other cases, effective interventions like student mentoring [14] are too costly and not easy to scale up.

To overcome these limitations, the authors of this paper have designed ASSIST (Automated Scenarios of Future Study Progress), an AI-supported smartphone application to provide personalized feedback, encourage the students' self-reflection, and foreground the role of their efforts in shaping their academic outcomes instead of steering them actively towards specific courses of action.[1] Thus, this application integrates the advantages of low-threshold and scalable interventions (see, e.g., [9]) in an intelligent student monitoring system that illustrates absolute and relative performance metrics and provides success probabilities based on the student's decisions.

In addition, we expect to contribute to the empirical evidence on the effects of self-reflection on academic outcomes by evaluating the tool in an experimental setting. This intervention will be rolled out as a randomized controlled trial during the summer term of 2023. It will be followed by an assessment of the intervention's effects on the number of exam registrations, academic performance and success, and other observable student outcomes.[2] In this contribution, we present the application's technical characteristics and functionalities, including a data description, and the algorithmic approach to obtain success probabilities. Lastly, we describe the content and functionalities available in the final product.

2 Student-level data

To compute student success predictions, we use standardized administrative data collected between 2007 and 2023 by a large state university from the German state of North Rhine-Westphalia. In addition to demographic variables, information on university entrance qualifications and study progression is also available. In addition to these basic features, it is possible to generate additional variables through feature engineering or by linking some of our data to external databases. For instance, we can approximate the change in a student's social environment by estimating the distance between the district of high school graduation and the university.

These data are exported from the university information system via a unique and anonymized person ID[3] to prevent tracing to spe-

* Corresponding Author. Email: Rumert@wiwi.uni-wuppertal.de

[1] Some studies show that promoting self-reflection through goal-setting (e.g., [15]) and self-monitoring (see [10]) in the context of higher education can benefit students.

[2] A quality assurance process has been implemented throughout this application's development, including pre-testing and user interviews.

[3] Our data contain 68,124 unique IDs from all undergraduate programs, after removing exchange students and visiting students who are only temporarily

cific students during data processing.[4] It is also necessary to distinguish between current students, graduates, and dropouts, which is not always straightforward. In this context, we consider that a *student* is a person who is continuously enrolled, i.e., less than one semester of interruption; *graduation* is defined as the successful completion of a study program; and the status *dropout* is assigned if a student is disenrolled from the university for one or more semesters without finishing the final exam. Data on university examination results can only be linked to the unique ID of a student and not to their respective study program preventing a clear comparison of academic achievements and activity within and across different fields of study.[5]

Lastly, we match examination data collected by the university with our definition of a student/observation. Then, we check the date and the assigned person ID of every exam and assign them based on the correct ID and a continuous enrollment time frame. This matching procedure is necessary since we define re-enrollment after a one-semester disenrollment as a new observation. Suppose a student receives credit points for passed courses in the past, e.g., from other universities or a past period of study. In that case, the credit points will be assigned to the semester that corresponds with the date documented by the university administration, which is usually the first semester.

3 Student success predictions

To estimate probabilities for active students in varying semesters, we censor our training data by only including former students that reached at least the semester of interest, excluding variables that change over time, and excluding variables that include information on later semesters. This results in one training dataset for each prediction setting (e.g., after semester 1).[6] Now, the current probability of graduation can be estimated and used as a starting point. To estimate the probability of success in the upcoming semester (depending on potential future students' performance) for every active student, we multiply our observations until the prediction dataset includes one version of every student for every possible outcome. Afterward, we can predict the probability of graduating for these synthetical observations and receive the full spectrum of approximations.

To show how well each student performed in passed credit points and achieved average grades compared to their peers in the past semesters, we group the relevant students of each comparison group and estimate deciles for outcomes of interest. The highest surpassed level will be saved as the minimum number of peers the student surpassed in the respective semester. The peer group is defined as the same cohort of students, i.e., the same enrollment date, enrolled at the same faculty in the respective semester.[7]

enrolled.

[4] Data pre-processing includes cleaning, imputation of missing values, and transformation. Depending on the logical structure of the variable, missing values are filled with a zero or the mean value. Numeric variables are normalized, and categorical variables are transformed into a binary structure.

[5] Ungraded examinations like internships and some practical assignments are only considered for estimating variables that do not depend on this information, e.g., the number of passed exams.

[6] Our training datasets consist of observations of former students (graduates or dropouts) from 2008 to 2018.

[7] We save our estimates in multiple datasets, which can be linked using an identifier. A randomized and unique UUID replaces the person ID to ensure the anonymization of the results. In the last step, we write the data to a PostgreSQL database on a university server.

4 Algorithmic approach

The predictive model is a multilayer perceptron classifier (MLP) with two hidden layers $(16, 8)$, the logistic activation function, and 400 epochs for the stochastic optimization procedure and was implemented in Python with the `scikit-learn` library version 1.17.1. Aside from the mentioned parameters, we use the default settings suggested by `scikit-learn`. The model is utilized in a supervised learning approach to learn the non-linear relationship between our input features and an outcome of interest, given a training dataset with an arbitrary number of features and a target variable (here, the outcome of studying). After training the model, it can predict outcome labels or approximate a probability for each possible outcome label.

The outcome probability is the result of the input variables of an active student running through the hidden layers and its activation functions, plus a normalization layer with a softmax function. If one wants to predict labels, the highest estimated probability determines the label. In the here presented case, the last step is renounced since the probability of the label is of interest. To evaluate the model and address potential miscalibration issues [6], calibration curves for all semesters were estimated using students from cohorts between 2009 and 2013 for the training process and then using students from cohorts between 2014 and 2016 as the test dataset. Calibration curves plot the positive label's true frequency against the model's grouped predicted probability. Model calibration should be applied if our graph deviates too strongly from the perfectly calibrated line [12].

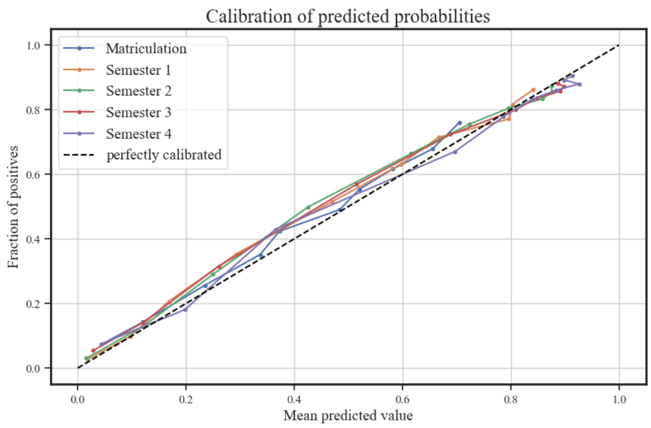

Figure 1: MLP calibration curves at enrollment and after the four following semesters

Figure 1 shows that for all displayed semesters and at matriculation, the MLP estimates reasonable success probabilities for the test cohorts. The model predicts higher graduation probabilities after university performance data is available, which resembles some level of uncertainty for the earliest prediction compared to the predictions of more advanced students. The positive deviation shows that our model slightly underpredicts the actual probability. To check if the model learns the changing parameters of interest correctly and yields an improved probability for graduation if a student passes more credit points or achieves a better average grade, partial dependence plots (PDP) [7] and individual conditional expectation plots (ICE) [5] were estimated. They visualize the response between an input feature and the target variable.

Figure 2 shows the PDP in orange, which represents the average change in the probability of graduation over passed credit points and

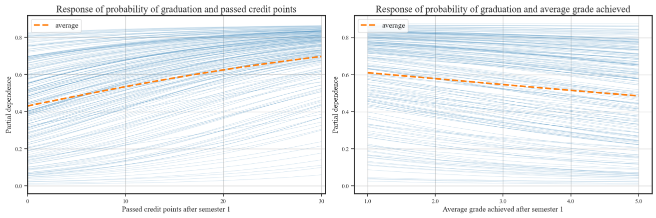

Figure 2: PDP and ICE for passed credit points and average achieved grade after semester 1

achieved average grade, and the ICE in blue, which represents the same change but for one individual observation after semester 1. The response rate for the later semester is similar. We can see that an increase in passed credit points leads to an increase in the probability of graduation. In contrast, a worse average grade leads to a decrease in the probability of graduation. This is what we expect to see, and both plots align with the correct change in outcome. The number of passed credit points expresses a higher response rate than the average grade, which we expect to see since credit points determine graduation.

To ensure that students observe an increase in the probability of graduation if they simulate a performance improvement, we filtered our database to check for reverse and implausible probability changes. The results show that all students observe an increase in their probability, and almost all students observe an increase of at least five percentage points if we compare a simulated result of 30 credit points and an average grade of 2.0 with a simulated result of 4 credit points and an average grade of 2.0. Most students experience an even greater improvement in their probability. We observe similar changes in the probability of graduation if we compare different average grade levels. The estimated probability of graduation for STEM students is the stickiest one. Their simulation results stay comparatively low, even if they show average to above-average performance results in their field. This makes sense since their academic workload is high, and the rigorous technical training runs through the entire course of study, with most STEM students graduating substantially later than the period of study suggested by the university.

We conducted ten interviews with one random student each to evaluate if the aggregated credit points displayed in the app and the estimated average grade were correct. Students reported that our estimations are approximately correct with minor deviations. These deviations emerge from how professors and their chairs handle grading and reporting academic results. Some start their seminars in the middle or relatively late into the semester, pushing deadlines for grade reporting in the early stages or sometimes even later into the upcoming semester. Some exams take place right before the end of the current semester, pushing grade reporting to the next semester.

5 Smartphone application

This smartphone application allows students to observe their performance and predict their success probability. While giving the students feedback on their current study progress, ASSIST also enables research on the effectiveness of performance feedback on study behavior. The front end was developed using Flutter, thus allowing deployment on Android and iOS. The smartphone application was realized in German, and its access is restricted to Bachelor students in the first four study semesters. The students log into the tool with the same user credentials required to use other digital services of the university.

Figure 3 shows the application view for a fictive student by utilizing dummy data. On the first page (left side) are static probabilities over the last semesters and a variable probability for the current semester. Students can use two sliders to select the average grade and the number of credit points for the current semester. The slider for the average grade includes all grade levels between 1.0 (highest grade) and 5.0 (fail), and the slider for the achieved credit points includes all possible credit points between 0 and 40. The lowest probability is achieved by adjusting the slider for the average grade to the lowest possible grade or adjusting the slider for the achieved credit points to the lowest possible credit points. The range of possible outcomes in the current semester is presented as a gray funnel, where the green line in the grey funnel shows the outcome, depending on the choice in the value of the sliders. The box above the slider contains information about the selected parameters and the associated probability of success.

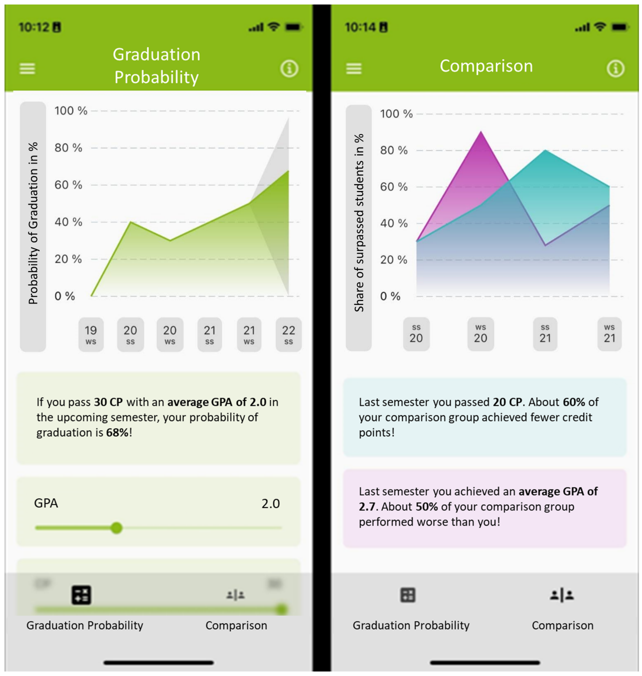

Figure 3: App showcase with dummy data

On the second page, students can compare their academic achievements to other students. The graph has two independent lines, colored by the matching text box below. They show what share of peers was outperformed regarding credit points and average grades. In the top right corner (below the power display), we implemented an information button, which opens a text box explaining the interpretability of predicted probabilities and the composition of the student's respective peer group. At the bottom of each page, there is a navigation bar.

At last, we implemented a questionnaire functionality to document the student's performance assessment before and after using the application. This allows the students to give feedback to, among others, their confidence in finishing their studies and their satisfaction with the studies themselves. The questions for the survey can be defined in the database and will be shown before giving the user access to the main navigation with the success prediction and the comparison page. The questionnaire can be deactivated and expanded.

Acknowledgements

The authors of this paper would like to thank the German Federal Ministry of Education and Research (BMBF) for the financial support within the project ASSIST with grant number 16DHB4001.

References

[1] Jaroslav Bayer, Hana Bydzovska, Tomáš Obšívač, Lubomír Popelínsk, and Jan Geryk, 'Predicting drop-out from social behaviour of students', in *International Educational Data Mining Society*, Chania, (2012).

[2] Johannes Berens, Kerstin Schneider, Simon Görtz, Simon Oster, and Julian Burghoff, 'Early Detection of Students at Risk-Predicting Student Dropouts Using Administrative Student Data and Machine Learning Methods', *Journal of Educational Data Mining*, **11**(3), 1–41, (2019).

[3] Raphael Brade, Oliver Himmler, and Robert Jäckle, 'Relative performance feedback and the effects of being above average — field experiment and replication', *Economics of Education Review*, **89**, 102268, (2022).

[4] Gerben W. Dekker, Mykola Pechenizkiy, and Jan M. Vleeshouwers, 'Predicting Students Drop Out: A Case Study', in *International Working Group on Educational Data Mining*, Cordoba, (2009).

[5] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin, 'Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation', *Journal of Computational and Graphical Statistics*, **24**(1), 44–65, (2015).

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger, 'On Calibration of Modern Neural Networks', in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, (2017).

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, New York, 2013.

[8] Ulrich Heublein, Christopher Hutzsch, and Robert Schmelzer, 'Die Entwicklung der Studienabbruchquoten in Deutschland', *DZHW Brief*, **5**, (2022).

[9] Oliver Himmler, Robert Jäckle, and Philipp Weinschenk, 'Soft Commitments, Reminders, and Academic Performance', *American Economic Journal: Applied Economics*, **11**(2), 114–142, (2019).

[10] Magdeleine D.N. Lew and Henk G. Schmidt, 'Self-reflection and academic performance: Is there a relationship?', *Advances in Health Sciences Education*, **16**(4), 529–545, (2011).

[11] Ioanna Lykourentzou, Ioannis Giannoukos, Vassilis Nikolopoulos, George Mpardis, and Vassili Loumos, 'Dropout prediction in e-learning courses through the combination of machine learning techniques', *Computers and Education*, **53**(3), 950–965, (2009).

[12] Alexandru Niculescu-Mizil and Rich Caruana, 'Predicting Good Probabilities With Supervised Learning', in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, (2005).

[13] Philip Oreopoulos and Uros Petronijevic, 'The Remarkable Unresponsiveness of College Students to Nudging And What We Can Learn from It', *NBER Working Papers*, (2019).

[14] Eline Sneyers and Kristof De Witte, 'Interventions in higher education and their effect on student success: a meta-analysis', *Educational Review*, **70**(2), 208–228, (2018).

[15] Cheryl J. Travers, Dominique Morisano, and Edwin A. Locke, 'Self-reflection, growth goals, and academic outcomes: A qualitative study', *British Journal of Educational Psychology*, **85**(2), 224–241, (2015).