Hungarian Educational Research Journal 11 (2021) 3, 297–310
DOI: 10.1556/063.2021.00017

AKADÉMIAI KIADÓ

Detecting at-risk students on Computer Science bachelor programs based on pre-enrollment characteristics

ERIKA B. VARGA* and ÁDÁM SÁTÁN

Institute of Information Science, University of Miskolc, Hungary

THEMATIC ARTICLE

Received: July 27, 2020 • Accepted: January 26, 2021
Published online: April 21, 2021

ABSTRACT

The purpose of this paper is to investigate the pre-enrollment attributes of first-year students at Computer Science BSc programs of the University of Miskolc, Hungary in order to find those that mostly contribute to failure on the Programming Basics first-semester course and, consequently to dropout. Our aim is to detect at-risk students early, so that we can offer appropriate mentorship program to them. The study is based on secondary school performance and first-semester Programming Basics course results from the last decade of over 500 students. Secondary school performance is characterized by the rank of the school, admission point score, and foreign language knowledge. The correlation of these data with the Programming Basics course result is measured. We have tested three hypotheses, and found that admission point score and school rank together have significant impact on the first-semester Programming Basics course results. The findings also support our assumption that students having weaknesses in all examined pre-enrollment attributes are subject to dropout. This paper presents our analysis on students' data and the method we used to determine the attributes that mostly affect dropout.

KEYWORDS

at risk students, potential dropouts, pre-enrollment characteristics, programming aptitude testing

[*] Corresponding author. E-mail: vargaerika@iit.uni-miskolc.hu

INTRODUCTION

In the Europe 2020 strategy, one of the goals is to have at least 40% of 30–34-year-olds complete higher education (Vossensteyn et al., 2015). This is in accordance with the global trends, since there is a world-wide observation that economy is best served by maximizing the level of education in the population (Yorke, 2004). For this reason national governments tend to widen participation in tertiary education. Thus, on the one hand, higher educational institutions need to accommodate large number of students with diverse social, economic and academic backgrounds. On the other hand, as a consequence of the policy aim of increasing participation rates, the risk of non-completion is increased (Yorke, 2004). In order to decrease dropout rates, universities need to take interventional steps to assure that the students remain in the system and achieve the learning outcomes required for completing a degree.

As one of the first measures, tertiary educators started classroom-based mentorship programs intended to help students in improving their educational outcomes, and thereby increasing student retention (Leidenfrost, Strassing, Schütz, Carbon, & Schabmann, 2014; Sandner, 2015; Fukaya et al., 2016; Hamilton, Boman, Rubin, & Sahota, 2019). Peer mentoring programs mainly target first-year students, because the transition from secondary school to university is a hard period, and vital in view of the student's decision to continue in higher education (Lunsford, Crisp, Dolan, & Wuetherick, 2017). Successful mentoring programs need a mass of mentors selected according to criteria such as high level of academic achievement, communication skills and conscientiousness; and also motivation of the mentees to spend extra time in the classroom. Since there is a heterogeneity in the backgrounds of university students, finding the most at-risk students is also crucial for the effort of cutting dropout rates down. The problem of dropping out of college before completing the bachelor degree has long been examined (Astin, 1964; Tinto, 1975; Pascarella & Terenzini, 1980). Most recent research focusing on the early detection of at-risk students propose various data mining models for predicting dropout (Dekker, Pechenizkiy, & Vleeshouwers, 2009; Kovacic, 2010; Shleena & Paul, 2015; Pradeep, Das, & Kizhekkethottam, 2015; Abu-Oda & El-Halees, 2015; Aulck, Velagapudi, Blumenstock, & West, 2016; von Hippel & Quezada-Hofflinger, 2017). For a review on mining in educational data, see (Dutt, Ismail, & Herawan, 2017; Baker & Inventado, 2014).

In accordance with the Europe 2020 strategy, in 2014 Hungary showed up 34.1% tertiary attainment in the target age group, which means an 8% increase compared to 2010. Since the policy of admission into higher education has changed over the last 3 decades, enrollment in tertiary education has nearly quadrupled since 1991. This has, however, not resulted in higher graduation rates. Less than half of all students are graduating within the required time, and the completion rate is one of the lowest in the OECD countries (OECD, 2017). The main reasons are insufficient academic preparation prior to enrollment and slow study progression (OECD, 2016).

In Hungary, Pusztai et al. (Pusztai, Fényes, Szigeti, & Pallay, 2019) studied student dropout scenarios, and created four student clusters on the basis of the identified factors. Researchers of Budapest University of Technology and Economics have worked out a method for predicting university dropout among STEM students (Nagy & Molontay, 2018), and created a web application that can be used as a decision support system for detecting at-risk students (Nagy, Molontay, & Szabó, 2019). In this model several pre-enrollment attributes are considered, but in (Horváth, Molontay, & Szabó, 2019) the authors argue against using the national secondary school ranking, because it does not take university performance of students into account. There are two factors why we cannot apply the above method:

- We do not possess the students' maturity exam data in the required details.
- Our university is regional, with heterogeneous student population, i.e., the range of admission points and secondary school ranks are great.

In this paper we are focusing on first-year students of Computer Science. This academic field is mostly open to, and suitable for experimenting with new approaches in teaching, such as multitasking and active learning during lectures, flipped classroom in practical courses, as well as problem-based and project-based learning (Carter, O'Grady, & Rosen, 2018). Students are highly motivated in getting a degree, nevertheless we still face the problem of high dropout rates at the University of Miskolc. We think this is mostly attributable to insufficient educational preparation, or the wrong choice of study field. Consequently we focus our efforts on early academic advising and selection of appropriate students for computer science.

The idea of constructing tests for selecting potential programmers dates back to the 1950s (Rowan, 1957; McNamara & Hughes, 1959; Stalnaker, 1965). Existing approaches to measuring aptitude for computer programming are partly based on psychological tests (Bishop-Clark & Wheeler, 1994; Wray, 2007) and specifically designed non-programming tasks (Tukianinen & Mönkkönen, 2002; University of Kent, 2020; AndroidApps, 2020). The combination of these tests can assess the applicant's logical reasoning, numerical problem solving and pattern recognition competencies, as well as the ability to follow complex procedures and attention to detail. In (Borzovs, Niedrite, & Solodovnikova, 2015), the authors argue for applying programming aptitude test to reduce attrition of first-year computer science students.

We have tested this hypothesis, and our research results show that a hybrid programming aptitude test alone is not a reliable predictor of success in a first-year programming course. Our students took this test – at the very beginning of their studies – with a rate of 75–80% on average, which may reflect our effective work in career orientation, or simply the "congenital" digital competencies of this generation which encourage the development of logical thinking and agility in solving problems. Despite the good test results, approx. 50% of our students fail to pass the first-year Programming Basics course exam, which drives us into further investigations. Following the methodology introduced in (Keane & Gray, 2019) we have analyzed the correlation between the pre-enrollment data of students and their first-year programming grade.

METHODOLOGY

The aim of this study was to investigate pre-enrollment characteristics of first-year students attending to Computer Science bachelor programs of the University of Miskolc in order to identify significant factors influencing their performance in the Programming Basics course. Three hypotheses were addressed:

H1 A programming aptitude test is a significant predictor of programming skills.

H2 Admission point score to university and the rank of the secondary school together are significant predictors of success in taking first-semester Programming exam.

H3 Skills in English as a foreign language have significant effect on programming skills.

Table 1. Pre-enrollment student characteristic data

Personal data	Secondary school characteristics	Secondary education performance data
Personal educational id	Secondary school identifier	Admission point score (APS)
Date of birth	City of secondary school	Study point (SP)
Gender		Maturity exam point (MP)
City of residence		Extra point (EP)
		Maturity exam subjects
		Maturity exam grades
		Maturity exam date
		Foreign language
		Language exam type
		Language exam level
		Language exam system
		Language exam date

Programming Basics course results were gathered from Neptun, the unified school management system of the University of Miskolc, for the academic years 2014–2018. In the case of students who took the course several times in the given period, the last grade was taken into account. Pre-enrollment educational data were provided by the Students' Center of the University of Miskolc for the same time period. Students' data include personal data, data about the secondary school and secondary education performance data, as listed in Table 1.

The original dataset contained 602 students' data. These included duplicates, since students may disenroll and later re-enroll for a study program; and also missing values. Duplicates were deleted, keeping only the last enrollment data for each student. Similarly, data records where we could not identify the secondary school, or could not calculate APS were deleted. Language exam was transformed into a binary variable: 1 denotes if language exam exists, 0 denotes if it is missing. The final dataset is comprised of 508 student records which is described by the statistics in Table 2.

Table 2. Statistical characteristics of the students' dataset

	Admission point score	Language certificate	Programming basics exam grade
Min	262	0	1
Max	473	1	5
Range	211	–	4
Mean	340.54	–	1.8324
Standard deviation	39.75	–	1.08
Mode	347	1	1
Median	340	1	1
Q1	309	0	1
Q3	364	1	2

For our study, we consider admission point score (APS) and language exam as best descriptors of pre-enrollment studies, because the feature selection process yields the highest values for these variables from the secondary education performance data group when the Programming Basics exam grade is the prediction variable. From students' personal data we have checked the relevance of gender and the year of birth by running Phyton's feature_selection.SelectKBest method in the scikit-learn package, but in both cases the contribution to Programming Basics exam grade is not significant.

We also searched for a representative descriptor of secondary school characteristics. For this reason, we have collected the ranks of the secondary schools of our students from *legjobbiskola.hu* (a website for national ranking of Hungarian secondary schools, accessed in January 2019). Here, secondary schools are grouped and ranked, on one hand, on the basis of the length of their study program (4, 6, or 8 years), and school type (grammar school or technical school), on the other. The ranking contains all together 1,645 educational programs ending up with maturity exam. Their ranking takes two factors into consideration: (i) results of national competency tests (measuring reading comprehension of language and mathematical text together with long-term concentration; completed in the sixth, eighth, and tenth grades), and (ii) results of maturity exams. In this website, Hungarian secondary schools are ranked according to the results of 5 consecutive years, so the collected data refer to 2014–2018.

BACKGROUND – HUNGARIAN EDUCATION SYSTEM

Measures of pre-enrollment studies

Admission to higher education institutions is based on the ranking of applying students according to their admission point score (APS), which can be calculated in two ways:

$$APS = SP + MP + EP \quad \text{or} \quad APS = 2 \cdot MP + EP$$

Study points (SP) can sum up to 200. Half of the points come from secondary school grades of the four core subjects (Mathematics, Hungarian language and literature, History, and a foreign language), plus a chosen science subject regarding the last two years, multiplied by 2. The other half is the average of the percental results of the five maturity exam subjects. Maturity exam in Hungary is the centralized final exam of secondary school studies. It consists of the four core subjects, plus a subject of arbitrary choice that has been being learned for at least 2 years.

Maturity exam points (MP) can also sum up to 200. This is computed as the sum of the percental results of two given maturity exam subjects, which are specified according to academic programs.

Students can choose to take the maturity exam of a subject at normal or at advanced level. Since this difference is not considered in MP, it is one chance to get 50 extra points (EP) per subject. EP may also include points for foreign language certificate, higher-level vocational training, prestigious place in sport, art or academic competitions, as well as points for equalizing opportunities; but their sum is maximized in 100 points.

Students of Computer Science Bsc programs

In Hungary, there are three types of BSc programs within computer science: Business Informatics, Computational Science and Computer Engineering. When calculating MP of students, the percental maturity exam results in Mathematics, and in Informatics or Physics (or in a corresponding vocational subject) are summed up. From 2020 one of these exams should be taken at advanced level.

Informatics is taught only in the first two years in secondary school (apart from specific secondary technical schools), focusing on the basic topics of computer architecture and use of office applications. Since the availability of optional supplementary courses is low, most students enter higher education without any programming skills.

Computer Science BSc programs are practice oriented (60% practical vs. 40% theoretical courses) where the first-year curriculum consists mainly of common fundamental subjects, such as Discrete Mathematics, Mathematical Analysis, Computer Architectures, Operating Systems and Computer Programming. Programming courses are designed so as to cover the basic concepts of procedural and object oriented programming, incorporate praxis and develop problem solving and algorithmic thinking. Programming knowledge is not a pre-requisite, but students need logical and abstract thinking skills in order to meet the requirements.

Tertiary educational institutes are not testing the students' level of these skills before enrollment. They rely on the maturity exam in Mathematics. As a consequence of high dropout rates, it is a trend now to prepare tests for freshmen for early detection of at-risk students. These tests are for evaluating the competencies in the use of mathematical tools.

At the University of Miskolc, we introduced the application of a hybrid programming aptitude test as well, that involves logical reasoning, numerical problem solving, pattern recognition (non-verbal reasoning) and verbal reasoning. Our experiences are summarized in the next section.

APPLICATION OF PROGRAMMING APTITUDE TEST

For testing the H1 hypothesis, we have created a programming aptitude test (PA test) for evaluating the freshmen computer science students' logical reasoning, numerical problem solving and pattern recognition competencies, as well as the ability to follow complex procedures and attention to detail based on the non-programming problem types used in (Mikova & Hulkova, 2013; University of Kent, 2020).

The test includes 40 questions for 50 minutes (available in Hungarian at *users.iit.uni-miskolc.hu/~vargae/Segedlet/ProgrammingAptitudeTest.pdf*). We applied the voluntary test in two consecutive years: in 2017, 145 students took the test (from 170 first-year students) with a rate of 80.65% on average (mean 80.65%, std. deviation 10.2%); and in 2018, 176 students took the test (from 198 first-year students) with a rate of 75% on average (mean 75.085%, std. deviation 11.2%). These results were promising, but it turned out that they do not have enough predictive power. Table 3 shows how these students performed at the Programming Basics (PB) course exam at the end of the first semester.

The distribution of students in the PA test result groups follows a bell curve in both research periods. In 2017, 47% while in 2018, 63.63% of test taker students failed on the Programming Basics exam. The standard deviation of failure percentages is small, if the 0-failure groups are not considered. From the diagrams of Fig. 1 we can conclude that only the test results approximating or reaching 100% guaranteed success in passing the Programming Basics course exam. In all the other test result groups, the ratio of failing students was above 40%.

Table 3. Correlation between Programming Aptitude (PA) test results and Programming Basics (PB) course failure

Test group	PA test result	Num. of students	Percent of test taking students	Num. of PB course failing students	Percent of failing students in test group
2017 (Number of test takers: 145)					
1	[52.5%, 60%)	4	2.75%	2	50.00%
2	[60%, 70%)	17	11.72%	10	58.82%
3	[70%, 80%)	35	24.14%	17	48.57%
4	[80%, 90%)	54	37.24%	22	40.74%
5	[90%, 95%)	24	16.55%	10	41.66%
6	[95%, 99%]	9	6.21%	6	66.66%
7	100%	1	0.69%	0	0.00%
2018 (Number of test takers: 176)					
1	[43%, 60%)	16	9.10%	12	75.00%
2	[60%, 70%)	32	18.18%	24	75.00%
3	[70%, 80%)	63	35.80%	39	62.00%
4	[80%, 90%)	47	26.70%	31	65.96%
5	[90%, 95%)	7	3.97%	6	85.71%
6	[95%, 99%]	9	5.11%	0	0.00%
7	100%	2	1.14%	0	0.00%

For this analysis, the Programming Basics exam grade variable was transformed into a binary variable: 1 denotes successful, 0 denotes unsuccessful exam. Point biserial correlation coefficient is calculated between PA test result percentages and PB exam success indices. In 2017 this coefficient is 0.08 (P-value = 0.33), while it is 0.02 in 2018 (P-value = 0.76). Thus, although the point biserial correlation coefficient between the two variables is slightly positive in both years, it turns out to not be a statistically significant correlation. This tells us that PA tests comprising only non-programming tasks cannot be used to predict success or potential failure in the first semester of Computer Science studies, hence H1 is rejected.

DETECTING AT-RISK STUDENTS

Henceforth we measure programming aptitude by the Programming Basics first-semester course exam results (y). We can rely on the outcome of this subject, because more than 80% of students completing the first semester will finally get a degree, based on the data registered in the unified school management system of the University of Miskolc. The research question is how this outcome relates to overall secondary school performance.

Between APS and school rankings there is a slightly negative linear correlation ($r_{x1,x2} = -0.2$, P-value = 0.00000002). From Fig. 2, that displays APS in function of school rankings, we can see that students can achieve high final scores in more prestigious, top-ranked schools, and also in schools having worse ranks. This figure reflects the heterogeneity of our student population: we have students coming from all types of schools (half of them come from the first tierce of schools in the national ranking) with APS ranging from 262 to 473 points. Concerning the overall population, the point biserial correlation coefficient between Programming Basics

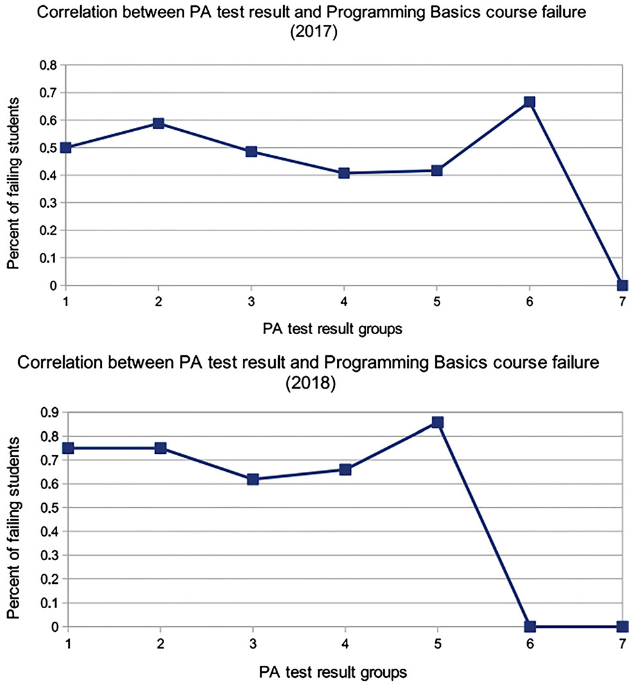

Fig. 1. Programming Basics course failure in function of Programming Aptitude (PA) test results

exam success and APS is 0.19 (*P*-value = 0.000006), which means that there is statistically significant correlation between the two variables. The point biserial correlation coefficient between Programming Basics exam success and school rank is −0.06 (*P*-value = 0.14), which cannot be considered as statistically significant correlation.

In order to reduce the population's heterogeneity, we have classified the students into four groups:

1. Students coming from top-ranked schools, having low APS.
2. Students coming from top-ranked schools, having high APS.
3. Students coming from low-ranked schools, having low APS.
4. Students coming from low-ranked schools, having high APS.

For this classification, we have defined what we mean by top school rank and high APS point. Since our aim is to measure programming aptitude, these definitions are based on the first-year Programming Basics course exam results. The critical APS point (334) is calculated as the lowest score above which 90% of our students successfully take the first-year Programming Basics course exam. The boundary ranking score between top-ranked and low-ranked schools (411) is calculated as the highest ranking below which 90% of our students successfully take the first-year Programming Basics course exam. The basic statistics and the correlation co-efficients for the four student groups are summarized in Table 4.

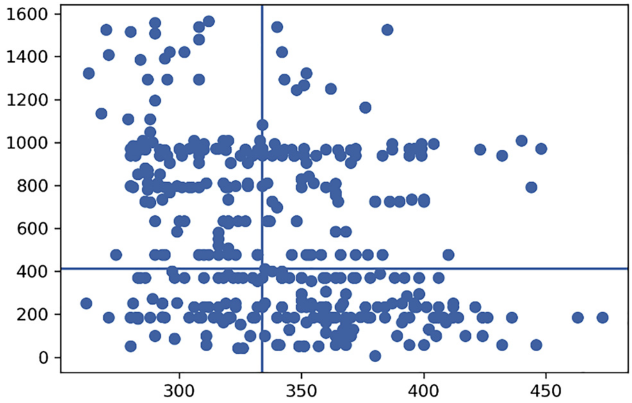

Fig. 2. Admission point scores of Computer Science students at the University of Miskolc (2014–2018) in function of the school rankings

Table 4. Statistical characteristics of student groups (x_1 – school rank, x_2 – APS, y – Programming Basics course result)

	Top-ranked school, low APS	Top-ranked school, high APS	Low-ranked school, low APS	Low-ranked school, high APS
Row count	69	175	148	116
min(x_1)	42	7	477	477
max(x_1)	400	411	1,565	1,539
median(x_1)	186	185	906	940
min(x_2)	262	334	263	334
max(x_2)	332	473	333	448
median(x_2)	309	360	304.5	352.5
$r_{y,x1}$ (P-value)	−0.08 (0.54)	−0.06 (0.44)	−0.01 (0.89)	−0.01 (0.92)
$r_{y,x2}$ (P-value)	0.11 (0.38)	0.10 (0.17)	0.13 (0.12)	0.12 (0.18)
$R_{y,\ (x1,x2)}$ (P-value)	0.12 (0.33)	0.38 (0.00000002)	0.14 (0.09)	0.32 (0.0005)

In H2 we implicitly assumed that neither APS, nor school ranking alone has significant effect on the Programming Basics course results, which is confirmed by the results in Table 4. Therefore we examined the multiple correlation coefficient, which takes both variables into consideration.

In the groups where students have high APS, secondary school rank and APS together explain 38 and 32% of the variance of Programming Basics course exam results, respectively. The P-value shows that the positive correlation coefficient value between the variables implies statistically significant correlation. Therefore we accept H2, i.e., APS and school rank together have significant effect on Programming Basics exam success, if a student has APS above 334.

Regarding the first-year Programming Basics course exam results, Fig. 3 shows that there are three groups where the likelihood of taking successful exam is higher than that of failing on the exam. The third group containing students from low-ranked schools and having low APS is the

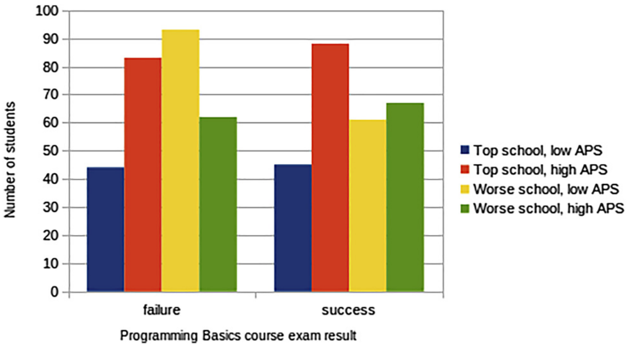

Fig. 3. Programming Basics course exam results in each of the student groups

most critical one. This class includes at-risk students who need extra help in completing this course.

The research published in (Soloway & Spohrer, 1989) shows that the skills in natural language have a great deal of impact on students' conceptions and misconceptions of programming. We have tested this hypothesis in H3 by taking into account the students' knowledge of a foreign language. 75% of our students have foreign language certificate, 95% of which certify intermediate or higher level knowledge of English. For the whole student population, this variable's correlation coefficient with Programming Basics exam success is 0.04 (P-value = 0.39). Since this is not considered as significant correlation, we examined the effect of having langauge certificate in each of the defined student groups. In the first two groups (students coming from top-ranked schools) this variable's bivariate correlation coefficient is $r_{y,x3} = 0.03$

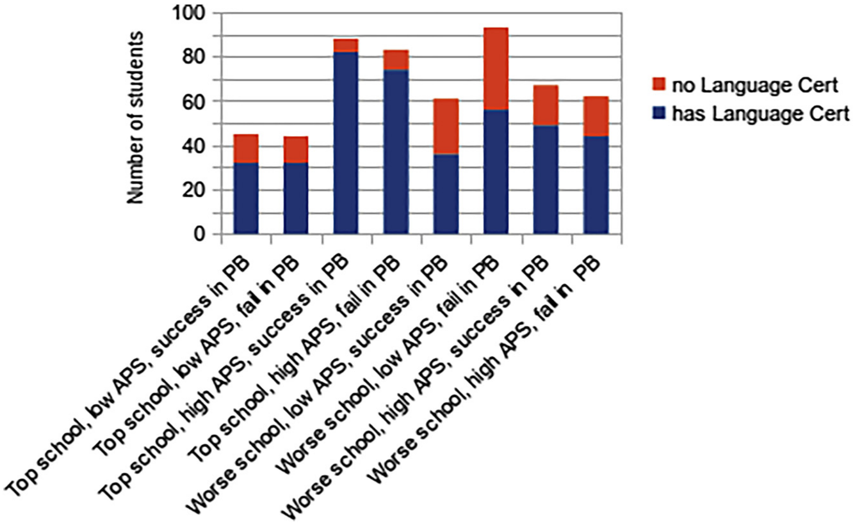

Fig. 4. The effect of foreign language certificate to Programming Basics course exam results

(P-value is 0.78 and 0.66, respectively), while in the second two groups (students coming from low-ranked schools) this value is $r_{y,x3} = 0.02$ (P-value is 0.81 and 0.84, respectively). Adding the language certificate variable to APS and school rank, the multiple correlation coefficient changes only slightly in all four groups, so we can conclude that foreign language skills do not have significant impact on the Programming Basics course results, and hence we reject H3.

Figure 4 shows the number of students having, or not having foreign language certificate in each group. We can find that, in the third group, less than 60% of at-risk students have a foreign language certificate. This rate is better in all other student groups. It can also be seen from the diagram that, in this group, those who do not have a foreign language certificate are more likely to fail the Programming Basics course exam.

CONCLUSION

The general tendency of high dropout rates in the first semester of Computer Science BSc programs affects also the University of Miskolc in Hungary. As an instant solution, we started classroom-based mentorship programs on a voluntary basis, but they did not yield the expected results.

In order to detect at-risk students at an early stage, we created and applied a programming aptitude test for evaluating the freshmen computer science students' logical reasoning, numerical problem solving and pattern recognition competencies based on non-programming problem types. In 2017, 145 students took the test with a rate of 80.65% on average; and in 2018, 176 students took the test with a rate of 75% on average. Figure 1 shows that only the test results approximating or reaching 100% guaranteed success in passing the first-semester Programming Basics course exam. In all the other test result groups, the ratio of failing students was above 40%. Consequently, we reject H1 according to which programming aptitude tests comprising only non-programming tasks can be used to predict success or potential failure in the first semester of Computer Science studies.

Next, we have dived into the secondary school results of novices. We considered admission point score (APS) as the general descriptor of pre-enrollment studies together with the secondary school's ranking. Figure 2 shows the heterogeneity of our student population. They are from all types of secondary schools regarding *legjobbiskola.hu,* the Hungarian national ranking scheme, and having APS ranging from 262 to 473 points.

In order to reduce heterogeneity, we have created four classes of students according to their secondary school ranking and APS. In each group, we have analyzed the first-year Programming Basics course exam results and tested the joint effect of APS and secondary school rank on it, and found that in the groups where students possess high APS the multiple correlation coefficient is statistically significant, therefore we accepted H2. On the other hand, we could not verify the impact of a foreign language certificate on programming skills, so we rejected H3.

It also turned out, that most at-risk students are coming with APS lower than 334 and from secondary schools with a rank worse than 411 (see Fig. 3). Taking also their foreign language acquisition skills into consideration, we can conclude that students who do not have language certificate in this group are more likely to fail the Programming Basics course exam (see Fig. 4).

Conflict of interest: The author declares no conflict of interest.

ACKNOWLEDGEMENTS

The described article was carried out as part of the EFOP-3.6.1-16-00011 “Younger and Renewing University Innovative Knowledge City institutional development of the University of Miskolc aiming at intelligent specialisation” project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

REFERENCES

Abu-Oda, G. S., & El-Halees, A. M. (2015). Data mining in higher education: University students dropout case study. *International Journal of Data Mining & Knowledge Management Process*, *5*(1).

Android Apps on Google Play. Aptitude and logical reasoning. Available: play.google.com/store/apps/details?id=com.madguy.aptitude.lr [Accessed: Feb. 26, 2020].

Astin, A. W. (1964). Personal and environmental factors associated with college dropouts among high aptitude students. *Journal of Educational Psychology*, *55*(4), 219–227. https://doi.org/10.1037/h0046924.

Aulck, L. S., Velagapudi, N., Blumenstock, J., & West, J. (2016). *Predicting student dropout in higher education*. arXiv preprint arXiv:1606.06364.

Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. Learning analytics. Springer, pp. 61–75.

Bishop-Clark, C., & Wheeler, D. D. (1994). The Myers-Briggs personality type and its relationship to computer programming. *Journal of Research on Computing in Education*, *26*(3), 358–370.

Borzovs, J., Niedrite, L., & Solodovnikova, D. (2015). *Computer programming aptitude test as a tool for reducing student attrition.* In *Proc.* of the 10th int. Scientific and practical conf. on environment, Technology. Resources (*Vol. III*, pp. 29–35). http://dx.doi.org/10.17770/etr2015vol3.175.

Carter, J., O’Grady, M., & Rosen, C. (Eds.) (2018). *Higher education computer science – a manual of practical approaches*. Springer, http://dx.doi.org/10.1007/978-3-319-98590-9 3.

Dekker, G. W., Pechenizkiy, M., & Vleeshouwers, J. M. (2009). Predicting students dropout: A case study, In *Proceedings of the 2nd International Conference on Educational Data Mining, EDM 2009*, July 1–3, 2009. Cordoba, Spain, pp. 41–50.

Dutt, A., Ismail, M. A., & Herawan, T. (2017). A systematic review on educational data mining. *IEEE Access* 5, 15991–16005, http://dx.doi.org/10.1109/ACCESS.2017.2654247.

Fukaya, T., Uesaka, Y., Tanaka, E., Shinogaya, K., Nishio, S., & Ichikawa, S. (2016). Effect of a high school peer-tutoring program on the quality of students’ interactions and learning strategy use. *The Japanese Journal of Educational Psychology*, *64*, pp. 88–104. http://dx.doi.org/10.5926/jjep.64.88.

Hamilton, L., Boman, J., Rubin, H., & Sahota, B. (2019). Examining the impact of a university mentorship program on student outcomes. *International Journal of Mentoring and Coaching in Education*, *8*(1), 19–36. Publisher: Emerald Publishing Limited, https://doi.org/10.1108/IJMCE-02-2018-0013.

von Hippel, P., & Hofflinger, A. (2017). The data revolution comes to higher education: Identifying students at risk of dropout in Chile (November 19, 2017). Available at SSRN: https://ssrn.com/abstract=3073912.

Horváth, N., Molontay, R., & Szabó, M. (2019). *Who are the most important “suppliers” for universities? Ranking secondary schools based on their students’ university performance*. In Proceedings of the 2nd Danube conference for higher education management.

Keane, M., & Gray, G. (2019). An investigation into the pre-enrolment characteristics of students to identify factors predictive of academic performance within first year computing and engineering programmes of study in a higher educational institution. *All Ireland Journal of Teaching and Learning in Higher Education* (AISHE-J), *11*(1).

Kovacic, Z. (2010). *Early prediction of student success: mining students enrolment data*. In Proceedings of informing science & IT education conference.

Leidenfrost, B., Strassing, B., Schütz, M., Carbon, C. C., & Schabmann, A. (2014). The impact of peer mentoring on mentee academic performance: is any mentoring style is better than no mentoring at all? *International Journal of Teaching and Learning in Higher Education*, *26*(1), 102–111, ISSN 1812-9129.

Lunsford, L. G., Crisp, G., Dolan, E. L., & Wuetherick, B. (2017). Mentoring in higher education. Ch.20. In *The SAGE handbook of mentoring*, SAGE Publications Ltd. https://doi.org/10.4135/9781526402011.n20.

McNamara, W. J., & Hughes, J. L. (1959). *Manual for the revised programmer aptitude test.* New York: International Business Machines Corporation.

Milkova, E., & Hulkova, A. (2013). Algorithmic and logical thinking development: Base of programming skills. *WSEAS Transactions on Computers*, *12*(2). E-ISSN: 2224–2872.

Nagy, M., & Molontay, R. (2018). Predicting dropout in higher education based on secondary school performance. In *IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, Las Palmas de Gran Canaria, Spain, 2018, pp. 000389–000394, https://doi.org/10.1109/INES.2018.8523888.

Nagy, M., Molontay, R., & Szabó, M. (2019). A web application for predicting academic performance and identifying the contributing factors. In *47th Annual Conference of SEFI*, 2019, pp. 1794–1806.

OECD (2016). *Economic Survey Hungary 2016*, OECD Publishing, Paris, https://doi.org/10.1787/eco surveys-hun-2016-en.

OECD (2017). *Supporting entrepreneurship and innovation in higher education in Hungary*, Ch.1. Overview of the Hungarian higher education system, OECD/European Union, ISBN 978-92-64-27334-4, https://doi.org/10.1787/9789264273344-en.

Pascarella, E., & Terenzini, P. (1980). Predicting freshman persistence and voluntary dropout decisions from a theoretical model. *The Journal of Higher Education*, *51*(1), 60–75. https://doi.org/10.2307/1981125.

Pradeep, A., Das, S., & Kizhekkethottam, J.J. (2015). *Students dropout factor prediction using EDM techniques.* In Soft-computing and networks security (ICSNS), International conference on. IEEE, pp.1-7.

Pusztai, G., Fényes, H., Szigeti, F., & Pallay, K. (2019). Dropped-out students and the decision to drop-out in Hungary. *Central European Journal of Educational Research*, *1*(1), 31–40. https://doi.org/10.37441/CEJER/2019/1/1/3341.

Rowan, T. C. (1957). Psychological tests and selection of computer programmers. *Journal of the Association for Computing Machinery*, *4*, 348–353.

Sandner, M. (2015). The effects of high-quality student mentoring. *Economics Letters*, *136*. 227–232. https://doi.org/10.1016/j.econlet.2015.09.043.

Shaleena, K., & Paul, S. (2015). *Data mining techniques for predicting student performance.* In Engineering and Technology (ICETECH), 2015 IEEE international conference on. IEEE, pp. 1–3.

Soloway, E., & Spohrer, J. C. (1989). Some difficulties of learning to program. In E. Soloway & J. C. Spohrer (Eds.), *Studying the novice programmer* (pp. 283–299). Hillsdale, NJ: LawrenceErlbaum Associates.

Stalnaker, A. W. (1965). *The Watson-Glaser critical thinking appraisal as a predictor of programming performance*. Proceedings of the Third Annual Computer Personnel Research Group, pp. 75–77.

Tinto, V. (1975). Dropout from higher education: a theoretical synthesis of recent research. *Review of Educational Research*, *45*(1), 89–125. Retrieved October 29, 2020, from http://www.jstor.org/stable/1170024.

Tukiainen, M., & Mönkkönen, E. (2002). *Programming aptitude testing as a prediction of learning to program.* In J. Kuljis, L. Baldwin, & R. Scoble (Eds.), Proc. PPIG *14*, pp. 45–57.

University of Kent. Computer programming aptitude test. Available: http://www.kent.ac.uk/careers/tests/computer-test.htm [Accessed: Feb. 26, 2020].

Vossensteyn, J.J., Kottmann, A., Jongbloed, B.W., Kaiser, F., Cremonini, L., Stensaker, B., et al. (2015). *Dropout and completion in higher education in Europe: Main report.*

Wray, S. (2007). *SQ minus EQ can predict programming aptitude.* Proceedings of the PPIG 19th Annual Workshop, pp. 243–254.

Yorke, M. (2004). *Leaving early: Undergraduate non-completion in higher education*. Routledge.