International Journal of Engineering Education Vol. 39, No. 4, pp. 836–844, 2023
Printed in Great Britain

0949-149X/91 $3.00+0.00

Predicting Academic Performance of Students in a Computer Programming Course using Data Mining*

IVAN PERAIĆ
Department of Information Sciences, University of Zadar, Croatia. E-mail: iperaic@unizd.hr

ANI GRUBIŠIĆ
Faculty of Science, University of Split, Croatia. E-mail: ani@pmfst.hr

The purpose of this work is to analyze data from the learning platform Moodle to predict the academic performance of the student in Programming class course. We used six machine learning classification techniques to extract a pattern from student Moodle using two datasets. In the first dataset, grade is in binary categorization (pass/fail), and in the second dataset grade is in a three-level categorization (fail, good and excellent). The research applies all possible combinations of eleven features for the selection of best predicting features, so we examined a total of 24432 prediction models on both datasets. The results show that Logistic Regression obtained the best results on binary dataset and Random Forests obtained the best results on three-level grade categorization, in terms of accuracy, precision and recall. We showed how the same classifiers on different features can give very similar results. In other words, there is no single best prediction model that significantly outperforms the others, but there are several very similar models that differ in the number of features selected and the selected classifier.

Keywords: data mining; predicting student performance; data mining classification; feature selection

1. Introduction

Achieving success in computer programming requires mastery of critical skills, such as problem-solving, computational thinking, critical thinking, and system design. Enhancing programming skills and promoting computational thinking has become crucial to equipping future citizens with the necessary expertise [1]. Therefore, programming has become an essential component in almost all engineering programs, emphasizing the need for a basic understanding of programming [2]. However, learning and teaching programming can be challenging, and novice programmers may quickly lose motivation and give up [3]. Despite significant differences between students and their learning environments, the approach to learning and teaching programming often remains unchanged. Traditional learning methods have limitations that can hinder the learning process, such as limited time and space to interact with each other, especially during the Covid-19 pandemic [4]. Given these circumstances, online learning has emerged as an alternative approach to support learning in its current state. Technological advancements have made it possible to simplify the learning process, increase flexibility, and enable learning to take place anywhere, anytime, and by anyone. Learning Management Systems (LMS) are widely used and produce huge amounts of data [5] and educational institutions are implementing new technologies to accumulate large amounts of data about students and the learning environment [6, 7]. One of the most commonly used LMS is Moodle (modular object oriented developmental learning environment), which is a free and open learning management system [8].

Predicting students academic performance is crucial for educational institutions seeking to enhance their students learning and achievement. However, this task is challenging due to the numerous factors influencing students course performance and learning. This study aims to predict students academic performance based on their programming activities in online learning. We focus on the engagement of the students by evaluating the number of their behaviors, and we make grade predictions at the end of the semester based on these activities in Computer Programming class. To achieve this, we employ the classification task with six classification algorithms: Logistic Regression (LR) [9], Support Vector Machines (SVM) [10], Random Forests (RF) [11], Neural Network (NN) [12], Naive bayes (NB) [13], and Decision Tree (DT) [14] . We used implementations of these algorithms from scikit-learn [15] and optimised model parameters for each algorithm by cross-validated grid-search over a parameter grid (sklearn.model selection.GridSearchCV). Predictions were compared in terms of classification accuracy rate, precision and recall.

In addition, this study investigates the effects of two different grade categorizations on data mining: three-level grade categorization and binary grade

* Accepted 6 March 2023.

categorization. Special attention is focused on feature selection for predicting the performance. Our dataset consists of eleven features, and we investigated which subset of features gives the best accuracy. The number of combinations according to our eleven features is 2036. Since we used six algorithms, we get 12216 prediction models (PMs). However, since we have two datasets that differ in the success variable, we examined a total of 24632 PMs. The code was written in Python, and the results are saved in a file. We discovered the best model by comparing the classification accuracy rate. According to this, the research questions can be stated as follows:

RQ1: How accurate are the proposed prediction models, in predicting students grade in a Computer Programming course at the end of the semester using interaction data from Moodle?
RQ2: What is the difference between two unlike grade categorizations in terms of predicting performance of students in a Programming course?
RQ3: What are the impacts of different feature selection on classification performance?

2. Related work on Predicting Student Academic Performance

Various authors have conducted studies to predict the academic performance of students using both supervised (classification) and unsupervised (clustering and association rule mining) educational data mining techniques [16]. Pal and Pal [17] analyzed students data and used three classification algorithms – ID3, C4.5 and Bagging to predict their academic performance. The analysis showed that ID3 had the highest accuracy rate of 78%, and the lowest average error rate was 0.16. Another study by Kabra and Bichkar [18] used decision trees to predict the performance of engineering students. The study showed that decision tree algorithms can be used to create a model that predicts the performance of engineering students in their first year based on their past academic performance. The true positive rate of the model for the "fail" class was 0.907, indicating that it successfully identified students who are likely to fail.

Yadav and Pal [19] analyzed three decision tree algorithms C4.5, ID3, and CART to predict the performance of engineering students in the final exam. The resulting decision tree provided insights into how many students were expected to pass, fail, or be promoted to the next academic year.

In a study by [20], the Support Vector Machine algorithm and the K-Nearest Neighbor algorithm were compared in terms of performance. The former produced slightly better results with a correlation coefficient of 0.96, highlighting its potential as an effective tool for predicting academic performance.

In another study by Evangelista [21], prediction models were developed using Moodle logs to predict students performance, including in the course "Intro to Computer Programming". The best features for prediction models were Activities Completed, Course Views, and Viewed in Mobile. In this study, the Random Forests algorithm had the highest accuracy at 90.9% for predicting whether a student will pass or fail a course. The main objective of study by [22] was to discover whether the students have used the LMS effectively to complete their learning process and enhance academic achievements in their study.

Rao and Kumar [23] presented a deep neural network model paper for predicting the students performance. It is the first time to use a deep neural network for the education data mining and predicting of students performance. The proposed model achieved an accuracy of 84.3%. This model can assist in predicting a student's academic performance and identifying students who are at a higher risk of failing, allowing for early intervention. Bergin et al. [24] investigated six machine learning algorithms (NB, SMO, LR, backpropagation, DT, 3-NN) for predicting programming success, using the predetermined factors. NB was found to have the highest prediction accuracy. In a recent study [25], a predictive model has been developed to forecast the final grades of students in introductory courses at an early stage of the semester. To compare the efficacy of different machine learning algorithms, the researchers have tested eleven algorithms in five categories including Bayes, Function, Lazy (IBK), Rules-Based and Decision Tree using WEKA software.

In a study by Khasanah et al. [26], it was found that the accuracy of predicting student performance can be improved by carefully selecting high-influence attributes. To achieve this, feature selection was performed before classification. The student data used in their study was obtained from the Department of Industrial Engineering at Universitas Islam Indonesia. The authors utilized Bayesian Network and Decision Tree algorithms for classifying and predicting student performance. In another study, presented in [27], an educational software was developed to predict student success in computer engineering at the University of Rwanda using variables such as mathematical background, programming aptitude, problem-solving skills, gender, prior experience, previous computer programming experience, and e-learning usage. The system collected input values and generated results using decision tree and Regression algorithms. In a study by Romero et al. [28], the effectiveness of various data mining techniques was compared for classifying engineering students based on their usage data in multiple Moodle courses at Cordoba University. The study utilized well-known classification methods, including statistical methods, decision trees, rule and fuzzy rule induction methods, and neural networks.

Manikandan and Chinnadurai [29] presents the use of TensorFlow for classifying and predicting students performance in academic and non-academic activities based on the analysis of 2500 data from Tamil Nadu. The study utilizes convolutional artificial neural network and TensorFlow entropy for pattern recognition, resulting in an accuracy factor of 75% to 85%. Romero and Ventura [30] used a Moodle mining tool to compare different data mining techniques for classifying students. They found that a good classifier model must be both accurate and understandable for instructors. Another study by [31] explained educational data mining with a case study on Moodle logs for visualization and classification purposes. In the study, five classification algorithms (DT, NB, SVM, RF, KNN) were presented to identify students performance in the early stages. Sixteen educational machine learning models were analyzed and compared in terms of their predictive power using LR [32]. The most effective individual predictors were found to be indicators composed of students quiz activities. However, including additional less effective predictors related to other activities could significantly improve the model's efficiency. The data from Moodle logs was also used to predict student success through analysis, as in the case study conducted by Ademi [8]. The main aim of Ademi's study was to predict students success and identify those who were at risk of dropping out. DT, Bayesian Network and Support Vector Machine algorithms were applied, and the results showed that DT had the best accuracy. Fuzzy-based approaches were also investigated for this problem. Palmer [33] explores the use of academic analytics to predict the academic performance of engineering students in their second year of study, using Logistic Regression and identifying significant predictor variables such as the mode of study, date of first login to LMS, and weighted average mark. The study shows that student data stored in institutional systems can be used to predict academic performance with reasonable accuracy and provides a simple but effective methodology for achieving this, with potential for targeted interventions to improve student success and retention outcomes. Hussain et al. [34] used Moodle log data to predict inactive and low-performing students during online learning. They employed FURIA, RF, and AIRS, and FURIA achieved the best results. Students introductory programming performance was predicted by [35] where Multilayer Perceptron, Naïve Bayes, SMO, J48 and REPTree were used on student related data to determine those students that may require additional support. SVM was utilized by [36] to construct a framework that can predict the success or failure of students in introductory programming courses.

To sum up, various researchers have investigated the problem of predicting student performance by employing a penty of data mining techniques. The results reveal a strong relationship between students activities and their academic achievements. However, a variation in research has been observed.

1. There is no consensus on the dataset used for predicting student performance. Many of these datasets include not only LMS interaction information but also other data (e.g., gender [16, 18–20, 25, 26, 33] , average self-study time [16], social parameters [16], psychological parameters [16], student's family size [16, 17], fathers and mothers qualification [16–20, 26] previous grades or academic performance [16–19, 26, 29, 33], civil status, city, income, enrollment year, major, programming experience, area of school, [16, 35, 19, 20]) in their prediction models without considering the fact that many of these variables fall outside the control of students and teachers alike. However, if dataset uses LMS data, there is lots of variations, from first login [33], total number of visits and clicks (on course [8, 31, 21, 36, 22] by night [21], by weekend [21]), assignements (done [28, 30, 25], submitted [28, 8], passed [21]), quiz (taken [8, 30, 25], passed [28, 30, 23], failed [28, 30], completed [21]), total time (on course [36, 33], on assignements, quizzes and forum [30], forum (creations [8], views [8], messages sent/read [28, 30, 33]), materials used [22].
2. Predicting student performance is common for all previously mentioned studies. However, student performance can be classified into different grade categorizations: binary grade categorization (pass/fail) [8, 16, 33, 32, 36], three-level grade categorization (fail, pass, good) [8, 30, 18, 19, 25, 28], four-level grade categorization [28, 17, 22], five-level grade categorization [8] and six-level grade categorization [34].
3. Predicting students performance can be related to one course [8, 16, 31, 34, 32, 22, 23, 18], or to many courses [22, 23, 17, 20, 25].
4. Different techniques and algorithms are used to predict performance, as previously mentioned. The choice of algorithm, as well as the number

of used algorithms, is at the discretion of the researcher.

5. The evaluation of prediction models is the last part of predicting student performance. Accuracy is the most intuitive performance measure, and it is simply a ratio of correctly predicted observations to the total observations. Accuracy is used for evaluation in [8, 16–20, 25, 31, 34, 32, 30, 36, 24, 26, 28, 29, 33]. Precision is used in [8, 16, 17, 31, 36, 22, 25], recall is used in [8, 16, 17, 25, 31, 36]. Some of studies used combination of accuracy, precision and recall to decide which model is more suitable to predict student the performance of the student [8, 16, 17, 25, 31].

3. Methodology

This section discusses the research methods followed in this research.

3.1 Data Collection

This study gathered data from students who enrolled in the "Introduction to Programming" course at the Faculty of Science, University of Split, Croatia. The data was collected through Moodle, which was used to distribute course material, lectures, homework, laboratory exercises, and quizzes. The data was collected over a period of fifteen weeks for three academic years: 2019–2020, 2020–2021, and 2021–2022. A total of 134, 145, and 178 students registered with Moodle for the first, second, and third year, respectively. A Moodle plug-in was used to automatically extract features representing students learning behaviors. The list of features is presented in Table 1. The final feature in the dataset is the categorical final grade, which reflects students performance and was added manually. This attribute is the target variable that we aim to predict. In the first dataset, grades were categorized as Pass or Fail (binary classification). We also tested the model using a three-level grade classification (Fail, Good, and Excellent) to compare its accuracy with different numbers of labels.

3.2 Preprocessing

It's unrealistic to expect a "perfect" dataset, so we began by removing incomplete results. We reviewed the dataset and removed incomplete results by eliminating entire students whose data was not complete. We identified variables with missing values (indicated as Null) and decided to remove morning1, morning2, and day2 due to incomplete data. Morning1 had 30 incomplete values, morning 2 had 20 incomplete values, and day2 had 15 incomplete values. We decided on elimination instead of filling incomplete values with potentially unreliable data. After removal, the dataset consisted of 12 variables and 427 student records.

3.3 Feature Selection

Feature selection involves selecting a relevant subset of features from a larger set that may contain irrelevant features [31]. This technique helps to analyze the relationship between independent features and the dependent feature by identifying the most influential independent features [37]. Feature selection algorithms can improve the performance of student performance PMs. The three main categories of feature selection algorithms are filter, wrapper, and hybrid models. Filter methods are performed during pre-processing and are not

Table 1. Features of the dataset

Feature	**Description**
logs	Total number of logins to the course
act1	Number of materials uploaded by students
act2	Number of materials viewed by the student
act3	Number of quizzes completed by the student
sumAct	Total number of activities (act1+act2+act3)
morning1	Number of logins from 00.01–06.00
morning2	Number of logins from 06.01–12.00
day1	Number of logins from 12.01–18.00
day2	Number of logins from 18.01–00.00
wlogs	Number of logins per week from Monday to Friday
weekendlogs	Number of logins during the weekend
timeOnExam	Time spent on the exam. We calculate it as a percentage of the total time spent in relation to the possible total time allowed
firstexam	Success in the first exam in percentage
attempts	Number of attempts while solving the exam. In Moodle, if specified, one exam can be solved more than once
grade	Final grades

Table 2. Number of combinations by selecting features

# of features	# of combinations
2	55
3	165
4	330
5	462
6	462
7	330
8	165
9	55
10	11
11	1

dependent on any learning algorithm, but instead rely on the overall features of the training data. Wrapper methods use learning algorithms to estimate the features, while hybrid feature selection combines the properties of both filter and wrapper methods [38].

In this study, we focused on every possible combination of features. Our dataset consists of 11 features, resulting in a total of 2,036 possible combinations (Table 2). This method is unique to our study, as feature selection is sometimes not mentioned in the literature, and researchers instead focus on final prediction and evaluation metrics. We implemented a total of 12,216 PMs by selecting six algorithms, and examined a total of 24,432 PMs since we have two datasets that differ in the success variable.

3.4 Evaluation Metrics

The code was written in Python, and the results were saved in a file. We determined the best model by comparing classification accuracy rate, precision, and recall. Accuracy is the most commonly used metric for evaluating the effectiveness of a prediction model [39]. It represents the ratio of correctly predicted results to total predictions. It is calculated according to the (1):

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{1}$$

where is:

True Positive (TP): the number of successful students who are correctly classified as "successful".
False Positive (FP): the number of successful students who are wrongly classified as "failed".
True Negative (TN): number of unsuccessful students correctly classified as "failed".
False Negative (FN): number of unsuccessful students who were wrongly classified as "successful".

Precision, recall, and F1 score are well-known performance indicators [40]. Precision represents the ratio of correctly predicted positive students (TP) to the total predicted positive students (TP + FP). It is calculated using (2):

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2}$$

Recall represents the ratio of correctly predicted positive observations to all observations. It is calculated using (3):

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

The results of each model are saved in an array that includes accuracy, features, and classification algorithms. The top ten results for each dataset were further compared in terms of precision and recall to determine the best prediction model.

4. Results

Two different grading systems, binary (Pass/Fail) and three-level grade categorization, were used to categorize the final grade attribute, resulting in two versions of the dataset. We compared the performance of six algorithms: Logistic Regression (LR), Support Vector Machines (SVM), Random Forests (RF), Neural Network (NN), Naive Bayes (NB), and Decision Tree (DT) on these datasets. Our goal was to identify which subset of features gave the best accuracy. Table 3 shows the performance of the top ten prediction models (PMs) on the first dataset, which used binary grade categorization. LR produced the best results for the binary grading version, achieving an accuracy rate of 82.17% on the testing dataset using act1, wlogs, attempts, and firstexam as independent features and grade as a dependent variable. Compared to other metrics, LR's precision and recall showed the highest results of 82.21% and 82.17%, respectively. The second and third best PMs were RF, with an accuracy of 82.17%, but with slightly lower precision and recall compared to LR. The second PM achieved a precision of 81.48% and a recall of 81.40% using logs, act3, sumAct, timeOnExam, day1, wlogs, and firstexam as independent features. The third PM achieved a precision of 81.24% and a recall of 81.32% using logs, act2, sumAct, weekendlogs, timeOnExam, day1, wlogs, attempts, and firstexam as independent features.

Additionally, seven more PMs had an accuracy of 81.39%, all using the Logistic Regression algorithm. These PMs had precision values ranging from 81.40% to 81.47% and recall values of 81.40% in every prediction model.

The performances of top ten PMs on three-level grade categorization dataset are shown in Table 4.

Table 3. Performance of 10 PMs on binary (Pass/Fail) dataset

PM1	LR	82.17%	82.21%	82.17%	**4**
PM2	RF	82.17%	81.48%	81.40%	7
PM3	RF	82.17%	81.24%	81.32%	9
PM4	LR	81.39%	81.47%	81.40%	4
PM5	LR	81.39%	81.40%	81.40%	4
PM6	LR	81.39%	81.40%	81.40%	4
PM7	LR	81.39%	81.40%	81.40%	6
PM8	LR	81.39%	81.40%	81.40%	5
PM9	LR	81.39%	81.40%	81.40%	5
PM10	LR	81.39%	81.40%	81.40%	5
	Algorithm	Accuracy	Precision	Recall	#Features

Table 4. Performance of 10 PMs on three-level grade categorization dataset

PM1	RF	65.62%	65.62%	64.80%	6
PM2	RF	64.8%	64.8%	64.8%	5
PM3	RF	64.8%	64.8%	64.8%	7
PM4	RF	64.06%	64.06%	63.78%	6
PM5	RF	64.06%	64.06%	63.78%	7
PM6	RF	63.28%	62.20%	63.30%	4
PM7	RF	63.28%	62.20%	63.30%	6
PM8	RF	63.28%	62.20%	63.30%	6
PM9	RF	63.28%	62.20%	63.30%	6
PM10	RF	63.28%	62.20%	63.30%	6
	Algorithm	Accuracy	Precision	Recall	#Features

When the grades are divided into three classes, the best PM's accuracy decreases to 65.62%, with a precision of 65.58% and a recall of 64.80%, based on the testing dataset with the following independent features: act1, act2, act3, timeOnExam, day1, and wlogs, and Grade as a dependent variable. All top ten PMs were obtained using RF. The second PM has an accuracy of 64.8%, precision 64.8% and recall 64.8% where features are act1, weekendlogs, timeOnExam, day1 and wlogs. The third PM has the same results as the second PM, but they differ in the set of features used. The third PM uses act1, act2, act3, timeOnExam, day1, wlogs, and attempts. The remaining top PMs differ in the set of features used for prediction, but their accuracy, precision, and recall results are the same. When comparing the top ten PMs, the differences are small between accuracy, precision, and recall.

5. Discussion

The research findings suggest that there is a difference in predicting student performance in a Computer Programming class course between the two grade categorization systems used in the study. The binary grade categorization system resulted in higher accuracy rates compared to the three-level grade categorization system. The LR model achieved the highest accuracy rate of 82.17% on the binary dataset, while the RFmodel achieved the highest accuracy rate of 65.62% on the three-level dataset. Additionally, the precision and recall scores were also higher for the binary dataset compared to the three-level dataset. These results suggest that the choice of grade categorization system can impact the accuracy of predicting student performance in a Computer Programming class course using interaction data from Moodle.

We found that by selecting different features, we achieved similar results in predicting student performance. In the two-level grade categorization dataset, the top ten prediction models had different feature sets but achieved comparable results. The accuracy score varied by only 0.78%, precision by 0.81%, and recall by 0.77%. Similarly, in the three-level grade categorization dataset, the top ten PMs achieved similar results with different sets of features, with the difference in precision ranging from 2.34% to 3.42% and recall from 1.5%. Interestingly, the number of features did not necessarily impact prediction power. We investigated the correlation between the features that produced the best results and found that the variables act3 and Attempts had the highest correlation coefficients of 0.578 and 0.568, respectively, in the binary dataset. In the three-level dataset, timeOnExam had a correlation coefficient of 0.027 and was part of the best-performing feature set for accuracy, precision, and recall. The inclusion of features with lower correlation coefficients, such as firstexam in the binary dataset, suggests that exploring all possible feature combinations may lead to better predictions.

We found that different sets of features can yield similar results in predicting student performance. For instance, on the two-level grade categorization dataset, the top ten prediction models achieved similar accuracy scores (within 0.78%), precision (within 0.81%), and recall (within 0.77%), even though they used different numbers of features (ranging from four to nine). Similarly, on the three-level grade categorization dataset, the top ten prediction models with different sets of features (ranging from four to six) also achieved similar results. However, there were some differences in precision (ranging from 2.34% between PM1 and PM10) and recall (ranging from 1.5%), indicating that some features may have more predictive power than others. Interestingly, the number of features selected did not always correlate with prediction power. We are interested in the correlation of the features that showed the best results. Therefore, we made correlations in the Python to check statistically how well the variables correlate with the success variable. According to the Table 5, on binary grade categorization dataset, we see that the variables act3 and Attempts have the highest correlation coefficient of 0.578 and 0.568 respectively. PM1 uses act1, wlogs, attempts and firstexam as fatures that gives best accuracy, precision, and recall. As we can see from Table 5 firstexam has a correlation of 0.281, therefore, we wonder whether we would have included this feature in the prediction set if we had not made all possible combinations, and it is in the best performing PM. Similarly, on three-level grade categorization dataset, the variable timeOnExam has a correlation of 0.027 (Table 6), and it is in the feature set that gives the best accuracy, precision, and recall.

Applying data mining techniques can be a valuable approach to accurately predict the final grades of Computer Programming class course students in binary grade categorization dataset, which can help educational institutions and instructors identify students who may need additional support and intervention to pass the course. Since Programming has become an essential component in almost all engineering programs, the results of this study can be extended to similar courses across all engineering programs as the ability to identify struggling students, as it can help prevent them from dropping out.

However, we acknowledge that this study has some limitations, such as the small sample size and the fact that the dataset was collected from only one institution. Future studies can expand the dataset to include more institutions and courses.

Table 5. Correlations with the success variable on binary grade categorization dataset

	Success
Success	1.000000
act3	0.577972
attempts	0.568133
sumAct	0.562287
act2	0.533918
wlogs	0.492447
day1	0.484030
weekendlogs	0.448967
firstexam	0.281024
timeOnExam	-0.367355
act1	-0.372585

Table 6. Correlations with the success variable on three-level grade categorization dataset

	Success
Success	1.000000
firstexam	0.259581
act1	0.216477
wlogs	0.068683
act3	0.054360
attempts	0.047969
sumAct	0.037362
timeOnExam	0.027363
act2	0.013527
weekendlogs	0.005869
day1	-0.026656

6. Conclusion

In this paper, we compared the performance of six Machine Learning algorithms in classifying students based on data from the Moodle system. Unlike other studies, we gave special attention not only to the Machine Learning algorithms but also to the selection of all possible combinations of features to achieve the best possible prediction. Our results showed that the same classifiers on different features can give very similar results. This means that there is no single best prediction model that outperforms the others, but several very similar models differing in the number of features selected and the classifier used.

We also investigated the effects of two different grade categorizations on data mining: three-level grade categorization and binary grade categorization. Our findings showed that the Logistic Regression algorithm was the best-performing algorithm, with an accuracy rate of 82.17% for the binary grade categorization and 65.62% for the three-level grade categorization. Although the binary dataset showed promising results, the model's accuracy at three levels was not very high. This indicates that predicting students final grades using their web usage data on a three-level grade categorization is a challenging task. Future studies can expand the dataset to include more institutions and courses, and consider additional offline features such as personal information, socio-economic status, study habits, motivation, and pre-university characteristics in addition to the online information used in this paper. It is crucial to note that obtaining these offline features is not as automated and effortless as acquiring the online information through Moodle. Instructors would need to manually provide the values for these new features.

This study highlights the potential of using data analytics and machine learning algorithms to improve student performance in Computer Programming class courses. The results suggest that careful selection of grading categorization and features can lead to accurate prediction of student performance, which can inform targeted interventions to improve student success and retention outcomes in computer engineering education, and can be further explored and applied in other areas of engineering education as well.

In our future experiments, we plan to apply feature selection methods to our dataset to determine the relevance of our best subset of features. We also aim to gather more data about students and incorporate larger datasets from similar courses and years to measure how offline and online data affect algorithm performance.

Acknowledgments – This work was supported by the Office of Naval Research grant, N00014-20-1-2066, Enhancing Adaptive Courseware based on Natural Language Processing.

References

1. T. Ball and B. Zorn, Teach foundational language principles, *Communications of the ACM*, **58**(5), 2015.
2. D. Topalli and N. E. Cagiltay, Improving programming skills in engineering education through problem-based game projects with Scratch, *Computers & Education*, **120**, pp. 64–74, 2018.
3. S. Mladenović , D. Krpan and M. Mladenović, Using Games to Help Novices Embrace Programming: From Elementary to Higher Education, *International Journal of Engineering Education,* **32**(1(B)), pp. 1–11, 2016.
4. D. Novaliendry, A. Huda, M. R. Cuhanazriansyah, H. K. Sani, H. Hendra and J. Karnando, E-Learning Based Web Programming Course in the COVID 19 Pandemic Time, *International Journal of Interactive Mobile Technologies (iJIM)*, **15**(20), pp. 117–130, 2021.
5. I. Kazanidis, S. Valsamidis, T. Theodosiou and S. Kontogiannis, Proposed framework for data mining in e-earning: The case of Open e-Class, u *International Conference of Applied Computing*, Rome, Italy, 2009.
6. D. Gaevi, S. Dawson and G. Siemens, Let's not forget: Learning analytics are about learning, *TechTrends*, **59**(1), pp. 64–71, 2015.
7. V. A. Nguyen, Q. B. Nqujen and V. T. Nqujen, A Model to Forecast Learning Outcomes for Students in Blended Learning Courses Based On Learning Analytics, *Association for Computing Machinery, ICSET*, 13–15 August 2018.
8. N. Ademi, S. Loshkovska and S. Kalajdziski, Prediction of Student Success Through Analysis of Moodle Logs: Case Study, *ICT Innovations 2019. Big Data Processing and Mining*, pp. 27–40, 2019.
9. D. G. Kleinbaum, Introduction to Logistic Regression, *Logistic Regression. Statistics in the Health Sciences*, New York, NY, Springer, pp. 1–38, 1994.
10. C. Cortes and V. Vapnik , Support-vector networks, *Machine Learning*, **20**, pp. 273–297, 1995.
11. V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan and B. P. Feuston, Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling, *Journal of Chemical Information and Computer Sciences*, pp. 1947–1958, 2003.
12. K. Gurney, An introduction to neural networks, London and New York: UCL Press Limited, 1997.
13. W. I. Geoffrey, Naïve Bayes, *Encyclopedia of Machine Learning*, Boston, Springer, 2011.
14. L. Rokach, *Data Mining with Decision Trees*, World Scientific, 2008.
15. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, **12**, pp. 2825–2830, 2011.
16. S. Verma, R. K. Yadav and K. Kholiya, Prediction of Academic Performance of Engineering Students by Using Data Mining Techniques, *International Journal of Information and Education Technology*, **12**(11), November 2022.
17. A. K. Pal and S. Pal, Analysis and mining of educational data for predicting the performance of students, *International Journal of Electronics Communication and Computer Engineering*, **4**(5), pp. 1560–1565, 2013.
18. R. R. Kabra and R. S. Bichkar, Performance Prediction of Engineering Students using Decision Trees, *International Journal of Computer Applications,* **36**(11), 2011.
19. S. K. Yadav and S. Pal, Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification, *World of Computer Science and Information Technology Journal (WCSIT)*, ISSN: 2221-0741, **2**(2), pp. 51–56, 2012.
20. H. Al-Shehri, A. Al-Qarni, L. Al-Saati, A. Batoaq, H. Badukhen, S. Alrashed, J. Alhiyafi and S. O., Student performance prediction using Support Vector Machine and K-Nearest Neighbor, *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada*, pp. 1–4, 2017.
21. E. D. Evangelista, Development of Machine Learning Models using Study Behavior Predictors of Students' Academic Performance Through Moodle, *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* ISSN: 2278-3075, **8**(6S3), April 2019.
22. M. Ayub, H. Toba, S. Yong and M. C. Wijanto, Modelling students programming subjects through educational data mining, *Global Journal of Engineering Education,* **19**(3), 2017.

23. M. G. Rao and K. K. Kumar, Students Performance Prediction in Online Courses Using Machine Learning Algorithms, *United International Journal for Research & Technology*, **02**(11), ISSN: 2582-6832, 2021.
24. S. Bergin, A. Mooney, J. Ghent and K. Quille, Using Machine Learning Techniques to Predict Introductory Programming Performance, *International Journal of Computer Science and Software Engineering (IJCSSE)*, **4**(12), December 2015.
25. I. Khan, A. Al Sadiri, A. R. Ahmad and N. Jabeur, Tracking Student Performance in Introductory Programming by Means of Machine Learning, *MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman*, pp. 1–6, 2019.
26. A. U. Khasanah and H. Harwati, A Comparative Study to Predict Student's Performance Using Educational Data Mining Techniques, *IOP Conference Series Materials Science and Engineering* **215**(1), p. 012036, 2017.
27. J. Batamuliza, Predicting success in computer engineering at university of Rwanda using machine learning, *Information Technologist (The),* **17**(1), 2020.
28. C. Romero, P. G. Espejo, A. Zafra, J. R. Romero and S. Ventura, Web Usage Mining for Predicting Final Marks of Students That Use Moodle Courses, *Computer Applications in Engineering Education,* **21**(1), 2013.
29. S. Manikandan and M. Chinnadurai, Evaluation of Students' Performance in Educational Sciences and Prediction of Future Development using TensorFlow, *International Journal of Engineering Education*, **36**(6), pp. 1783–1790, 2020.
30. C. Romero, S. Ventura, P. G. Espejo and C. Hervás, Data Mining Algorithms to Classify Students, *Educational Data Mining 2008, The 1st International Conference on Educational Data Mining, Montreal*, Québec, Canada, 2008.
31. M. Pokharel, Educational data mining in Moodle data, *International Journal of Informatics and Communication Technology (IJ-ICT)*, **10**(1), pp. 9–18, 2021.
32. L. Bognar and T. Fauszt, Different learning predictors and their effects for Moodle Machine Learning models, *11th IEEE International Conference on Cognitive Infocommunications – CogInfoCom 2020*, 2020.
33. S. Palmer, Modelling Engineering Student Academic Performance Using Academic Analytics, *International Journal of Engineering Education*, **29**(1), pp. 132–138, 2013.
34. M. Hussain, S. Hussain and W. Zhang, Mining Moodle Data to Detect the Inactive and Low performance Students during the Moodle Course, *ICBDR*, 27–29 October 2018.
35. M. Sivasakthi, Classification and prediction based data mining algorithms to predict students' introductory programming performance, *International Conference on Inventive Computing and Informatics (ICICI)*, pp. 346–350, 2017.
36. C. Van Petegem, L. Deconinck, D. Mourisse, R. Maertens, N. Strijbol, B. Dhoedt, B. De Wever, P. Dawyndt and B. Mesuere, Pass/fail prediction in programming courses, *Journal of Educational Computing Research*, June 2022.
37. A. Acharya and D. Sinha, Application of Feature Selection Methods in Educational Data Mining, *International Journal of Computer Applications*, **103**(2), pp. 34–38, 2014.
38. M. Zaffar, K. S. Savita, M. A. Hashmani and S. S. Hussain Rizvi, A Study of Feature Selection Algorithms for Predicting Students Academic Performance, *International Journal of Advanced Computer Science and Applications*, **9**(5), 2018.
39. A. Namoun and A. Alshanqiti, Predicting Student Performance Using Data Mining and Learning Analytics Techniques: A Systematic Literature Review, *Appl. Sci.,* **11**(237), 2021.
40. Á. M. Guerrero-Higueras, N. DeCastro-García and F. J. Rodriguez-Lera, Predicting academic success through students' interaction with Version Control Systems, *Open Comput. Sci.*, pp. 243–251, 2019.

Ivan Peraić is assistant at the University of Zadar, Department of Information Sciences, Croatia. He graduated from the University of Split, Faculty of Science in 2012. He is a PhD candidate on same faculty. Areas of scientific interest are learning analytics in e-learning systems and educational data mining. He is a member of the project "Adaptive Courseware based on Natural Language Processing (AC & NL Tutor)", funded by the Office of Naval Research, USA. He is an author and co-author of several papers.

Ani Grubišić is an Associate Professor at the University of Split, Faculty of Science. She graduated from the same Faculty in 2001, got her MS in 2007 and PhD in 2012 at the University of Zagreb, Faculty of Electrical Engineering and Computing. Areas of scientific interest are intelligent tutoring systems, adaptive courseware and learning analytics in e-learning systems. She is a Principal Investigator of two projects "Adaptive Courseware based on Natural Language Processing (AC & NL Tutor)", funded by the Office of Naval Research, USA. She is an author and co-author of more than thirty scientific papers.