Performance Indicators for Online Secondary Education: A Case Study

Pepijn van Diepen and Bert Bredeweg(✉)

Informatics Institute, Faculty of Science,
University of Amsterdam, Amsterdam, The Netherlands
PepijnVanDiepen@gmail.com, B.Bredeweg@uva.nl

Abstract. There is little consensus about what variables extracted from learner data are the most reliable indicators of learning performance. The aim of this study is to determine such indicators by taking a wide range of variables into consideration concerning overall learning activity and content processing. A genetic algorithm is used for the selection process and variables are evaluated based on their predictive power in a classification task. Variables extracted from exercise activities turn out to be most informative. Exercises designed to train students in understanding and applying material are found to be especially informative.

Keywords: Learning analytics · Learning performance indicators

1 Introduction

Learning Analytics (LA) provides insight into the progress of students and their learning performance. It analyses learner data with the aim to improve the learning process. Whereas the potential of the field is promising, results are still preliminary. A common approach is to let the prediction of learning performance act as guidance for teachers to identify students that need intervention. Quantitative data concerning *resource use*, *time spent on resources* and *grades* have been used for the prediction of learning performance [7,14]. However, confidence about what data are most suited is limited [1,14].

The aim of this study was to determine what aspects of learning behaviour can be extracted from the log-data of a Learning Management System (LMS) in secondary education and are reliable indicators of learning performance. An extensive set of potentially valuable variables was composed and several rounds of selection were applied in order to find the most informative indicators.

2 Related Work

2.1 Relevant Variables

Several studies indicated the statistical relevance of *resource usage* as predictive variable, often in terms of usage counts [7,9]. The *time spent* on learning objects

T. Bosse and B. Bredeweg (Eds.): BNAIC 2016, CCIS 765, pp. 169–177, 2017.
DOI: 10.1007/978-3-319-67468-1_12

(LOs) was also found to be an indicator of learning performance [8]. Variables concerning exercise behaviour such as the time spent on exercises, the number of successful and unsuccessful attempts, and scores were also reported to be related to learning performance [8,11,14]. Other studies found *study results* to be most informative [8,13], some reported *social interaction* being important [11], and numerous studies reported demographic data to be a reliable indicator [14,15].

The LMS used in our study offered a wide range of exercises and reading material. However, the inclusion of demographic data was prohibited due to privacy constraints and no data concerning social interaction was available.

2.2 Feature Selection Methods

When a large number of features is considered, a thorough feature selection process is essential to improve predictions, provide a deeper understanding of the case, guide the reduction of data, and yield simpler models [12]. A common initial means of feature ranking can be accomplished by analysing the Pearson correlation coefficients of features with the to be predicted variable. Univariate feature ranking can be preferable to multivariate feature selection methods due to its simplicity and scalability. However, features can be of more value when taken joined with other features. Univariate feature analysis does not detect such cases, Hence, multivariate feature selection methods should be considered [3]. Below the three main categories of feature selection algorithms are discussed.

Wrappers are simple yet robust feature subset selection methods that use prediction algorithm accuracies as a measure of subset quality. Wrappers search the entire space of feature subsets and therefore become computationally intractable when a large number of features is addressed. In order to cope with the scalability of wrapper methods, the search for feature subsets can be guided by search strategies such as Genetic Algorithms (GA). However, if wrapper methods are applied, the risk of overfitting increases.

Embedded methods are often faster solutions to feature selection since they embed the selection process into the training process and they use greedy search methods to address the problem of scalability. Greedy search strategies have the disadvantage that former decisions are never revisited, therefore they do not guarantee optimal solutions.

Filter methods are fast solutions to feature selection and are often used as a pre-processing step that uses general characteristics of the data to select features [12]. The advantage of filters is that the selection is made independent of the predictor that is used for the final prediction. Filter methods are often used for univariate feature analysis but in the context of multivariate feature analysis it is a reasonable approach to use a wrapper as a filter and train another often more complex predictor using the selected subset [3].

2.3 Prediction Methods

Numerous prediction algorithms have been used to classify learning performance on a *discrete* scale. Less research has been conducted concerning the prediction of learning performance on a *continuous* scale. Classification algorithms such as Decision Trees, Neural Networks, Naives Bayes, K-Nearest Neighbour, Support Vector Machines and Logistic Regression are regularly applied for LA [5,13]. Wolff et al. [15] used Decision Trees to predict whether a student would fail or pass a course. The accuracy of the predictive models varied from 0.77 to 0.98 over three different courses. This suggests that predictor performance could be course dependent. Macfadyen et al. [7] implemented a Binary Logistic Regression predictor in order to classify student failure. The classifier predicted student failure with an accuracy of 0.74. Minaei-Bidgoli et al. [8] used K-Nearest Neighbours and Decision Trees to classify student outcomes in terms of two and three learning performance classes. After the optimisation of algorithm parameters and by combining multiple classifiers an accuracy of 0.94 and 0.72 was achieved for the two- and three-classes respectively.

One could argue that simple prediction algorithms are preferred over more complex algorithms. This is because the decision making of simpler algorithms can be analysed better. These findings indicate that simple prediction algorithms such as Decision Trees and K-Nearest Neighbours can be successful in predicting learning performance.

3 Method

3.1 Data, Participants and Context

The data for the research presented in this paper was provided by educational publisher ThiemeMeulenhoff and was extracted from the logs of the online geography course De Geo[1]. De Geo is a geography course offering 1,166 exercises, 476 self-assessment tests and 9 chapters of reading material (i.e., the equivalent of a year of school material). The data consisted of chronological click logs (two months of data) and exercise results (7 months of data). The two datasets were combined to explore their full potential. Exercise data included the final score and a label stating whether the exercise was completed, incomplete or skipped.

The dataset included data of 226 first year, secondary education students from the Netherlands, aged 11–12. The course material included reading material (also referred to as *theory*), online exercises, and self-assessment tests. Each exercise was categorised according to Bloom's taxonomy for learning objectives [6]. Hence, 6 categories (*Remember* (89, 8%), *Understand* (139, 12%), *Apply* (676, 58%), *Analyse* (172, 15%), *Evaluate* (31, 3%) and *Create* (59, 5%)) which were hierarchically structured, meaning the mastery of the next category is supposed to follow from the mastery of the prior category. Exercise activity was analysed separately for each category.

Since all data was anonymous and no final grades were made available due to privacy constraints, learning performance had to be determined based on alternative sources. The self-assessment tests were designed to provide the students an

[1] https://www.thiememeulenhoff.nl/voortgezet-onderwijs/mens-en-maatschappij/aardrijkskunde/de-geo-onderbouw-9e-editie.

indication of their learning performance, therefore results on self-assessment tests were considered to be the most appropriate measure of learning performance. All students were labeled with the mean of their results on all self-assessment tests that they completed.

Due to the same privacy constraints the dataset is not publicly available.

3.2 Variable Selection

Composing an Initial Set. First, variables concerning overall online activity were considered (e.g., number of clicks, time online, theory/exercise time distribution). These variables were extracted from the data that was collected over all content together instead of specific types of content. Subsequently, content specific variables extracted from reading and exercise activities were considered. All data was categorised in terms of (*i*) exercise processing, (*ii*) theory processing, and (*iii*) overall behaviour. A set of variables was composed for each category based on the type of variables that were found to be reliable in the reviewed literature. In the case of the variables concerning exercise behaviour the data of each set of exercises belonging to a particular category was analysed individually. Additionally, all exercises were analysed when taken together as well. Two extraction methods were applied in order to address potential differences in difficulty between exercises. Method **A** assumes all exercises to be of equal difficulty and evenly time consuming, whereas method **B** does not. Method **B** compared and analyzed students' data per separate exercise while method **A** compared accumulated results per category.

Selection. Initially, a wide range of variables was included, followed by removing redundant and irrelevant variables from the set. A selection was made using a univariate variable selection method based on Pearson correlation with learning performance. All features that did not significantly correlate (p-value < 0.05) were discarded. Subsequently, multivariate variable selection was applied on the remaining variables. Embedded selection methods were not used since they rely on greedy selection algorithms which could exclude valuable features early in the process. Due to their computational complexity, wrappers based on the brute force methodology were also rejected. Therefore a combined filter/wrapper method as described in Sect. 2.2 was applied. A GA was used to guide the search for the best combination of variables[2]. GAs can be described as guided random search techniques that mimic the theory of evolution. They create populations of random individuals and select the best individuals to create the next population until an (sub)optimal solution is found. By using the prediction performance as fitness and variable subsets as individuals, GAs aim to select the strongest combination of variables.

For prediction a simple linear prediction model was selected as suggested by Guyon et al. [3]. The Linear Discriminant Analysis Classifier implemented by the Scikit Learn library [10] was selected for this purpose due to its simplicity and

[2] GA implementation from the DEAP library for evolutionary algorithms [2] was used.

low computational costs. The predictions were evaluated by a repeated 10-fold cross validation using 50 repetitions (see Sect. 3.3 for further explanation of these design choices). The algorithm was implemented following the guidelines provided by Fortin et al. [2]. The population consisted of 25 individuals, each representing a single feature subset. In each iteration of the evolutionary loop new offspring was generated by either mixing individuals using a uniform crossover method, mutating a single individual, or reproduction. As suggested by Fortin et al. the probability of mating individuals was set to 0.5 and the probability of mutating to 0.1. The mating of two individuals was accomplished using the uniform crossover method that exchanges the attributes of two individuals with an independent probability of 0.1. When an individual was mutated each feature was turned on or off with an independent probability of 0.05. Subsequently the offspring was joined with the original population and a selection made from the conjunction using the NSGA-2 selection operator provided by the DEAP library. The evolutionary loop was stopped at 75 iterations since that was the average point of convergence (stabilization of the population) from the test runs.

Since the feature subset space was searched extensively there was a significant probability that a combination of features was found that produces high predictive accuracy on the train set but would generalise poorly. Even when the fitness of individuals was determined by a cross-validation metric some overfitting could leak into the model. Since the dataset size was limited, and did not allow for a test set to be separated, 10-fold cross validation was applied to the GA feature selection process. The dataset was randomly split into ten parts, in a stratified fashion. After each iteration of the GA, the performance of the generated feature subset was tested on the validation set. The feature subset was optimised for the other 9/10 of the data and never saw the data in the validation set. Each fold had an optimal variable subset as output and a voting mechanism was used to make the final selection.

3.3 Classification

The predictive power of the selected variables was evaluated in two learning performance classification tasks: fail/pass and fail/sufficient/excellent. Classification algorithms such as Support Vector Machine (SVM), Gaussian Naive Bayes (GNB) and K-Nearest Neighbour (KNN) provided by the Scikit-Learn library [10] were used. All classifications were evaluated using repeated 10-fold cross validation. The k-fold Cross Validation (CV) estimator is a widely accepted model evaluation technique in the field of machine learning. Whereas it often produces unbiased estimates, the estimates can be highly variable when applied to a small dataset. Kim et al. [4] compared several bootstrap techniques to a repeated k-fold CV technique in order to address the problem of high variance in small datasets. They concluded that the repeated k-fold CV estimator outperformed bootstrap methods and recommended it for general use. Therefore the evaluation of all predictive models was conducted using a repeated k-fold CV, using $k = 10$ to maintain low bias. The number of repetitions was set to 50 because the model's confidence level stabilized at that point.

To find the optimal parameters for each learning algorithm the built-in grid search for optimal parameters of the scikit-learn library was used. It applies an exhaustive search trough a set of parameter options provided by the user to find the optimal parameters for the classifier.

All classifiers were evaluated in terms of accuracy, F2-score and recall of the class that represented the low performing students. Finally, a baseline classifier that was set to always predict the most common class was included in the evaluation.

4 Results

Over all variable categories together a total of almost 50 variables were considered. Within each category, variables concerning a student's time distribution, number of clicks and variations on those (such as ratios) were considered. Because variables extracted from different levels from Bloom's taxonomy were treated individually most variables originated from exercise behaviour. The selection process yielded a final set of 15 variables including almost all categories (Table 1). Two variables belonged to the overall activity category: total_clicks and the theory_exercise_ratio. Two variables belonged to the reading activity category: the theory_look_ups and the theory_look_ups_time. Eleven variables originated from the exercise activities, especially exercises from the *apply* category (five variables) and *understand* category (three variables) from Bloom's taxonomy were found to be reliable indicators. Notice that the variable exercise_incomplete_apply occurs twice in the list, once extracted using method **A** and once with method **B**. No variables came from the *remember* and *analyze* category. One variable came from both the *evaluate* and *create* category. From the exercise processing variables, the number of incomplete (wrong answer provided) was most informative, followed by the mean and total time spent on exercises. The last variable in the list concerned the mean time spent on an exercise over all of Bloom's categories together. To evaluate the predictive value of these variables, they were tested in two classification tasks. The baseline classifiers achieved an accuracy of 0.51 and 0.50 in the two and three class classification task respectively. The SVM classifier predicted most accurately for both classification tasks followed by GNB and KNN. An accuracy of 0.80 and recall of 0.84 of the *fail* class was achieved for the classification of two classes. For the classification of three classes an accuracy of 0.67 and recall of 0.67 was achieved. Other classification algorithms were also evaluated but resulted in less accurate predictions. However, all classifiers did perform significantly better ($p < 0.05$) than the baseline classifiers.

Table 1. Final selection of variables. Pearson correlation (r) with the learning outcome and corresponding p-values (p) are shown alongside with their ranking according to the GA (*votes*). Variable names end with A or B depending on the extraction method used.

#	Variable	Description	*votes*	r	p
1	exercise_incomplete_apply_A	Number of wrong answers on apply exercises	10	−0.52	0.00
2	theory_exercise_ratio	The ratio of time spent on exercises and theory	9	0.27	0.00
3	exercise_avg_time_understand_B	The mean time spent on understand exercises	9	0.23	0.00
4	exercise_time_evaluate_A	Total time spent on evaluate exercises	8	−0.16	0.04
5	exercise_completed_apply_A	Number of correct answers on apply exercises	8	0.16	0.04
6	theory_look_ups	Total number of theory look -ups during exercises	7	0.35	0.00
7	exercise_incomplete_understand_A	Number of wrong answers on understand exercises	7	−0.37	0.00
8	exercise_incomplete_apply_B	Number of wrong answers on apply exercises	7	−0.26	0.00
9	exercise_time_understand_A	Total time spent on understand exercises	6	0.23	0.00
10	total_clicks	Total number of clicks	6	0.17	0.03
11	exercise_avg_time_apply_A	The mean time spent on apply exercises	6	0.29	0.00
12	exercise_skipped_create_A	Number of skipped create exercises	6	−0.20	0.00
13	exercise_time_apply_A	Total time spent on apply exercises	5	0.24	0.00
14	exercise_time_A	Total time spent on exercises	5	0.20	0.00
15	theory_look_ups_time	Total time spent on theory look-ups	5	0.37	0.00

5 Conclusion

The aim of this study was to determine what LMS data best explains students' learning performance. In correspondence with the findings of Tempelaar et al. [14] the indicators concerning exercise processing were found to be most reliable. Variables extracted from exercise activities, that were designed to train students in *understanding* and *applying* material, were found to be especially informative. Both method **A** and **B** can be used to extract the variables although in general it seems that method **A** is sufficient. In contrast to Wolff et al. [15] variables describing general learning behaviour did contribute predictive value. Only theory processing variables related to exercise activity (look-ups during exercises) were part of the final variable list. This suggests that reading behaviour does not reveal much about learning outcome. However, a combination of features concerning overall activity, theory- and exercise-processing was needed to achieve the best prediction results. Therefore it is important to capture as many aspects of the learning process as possible in order to make accurate predictions.

To make predictions valuable for education, they need to be used to deliver valuable feedback to students. In our study none of the predictive models were analyzed in order to understand their decision making. In future research a better understanding of the predictions should be investigated and obtained.

Acknowledgements. The authors would like to thank ThiemeMeulenhoff for providing the resources for this study. Special thanks go to Joost Borsboom, Gilian Halewijn, Wouter van Rennes, Emiel Ubink and Johan Verhaar.

References

1. Esmeijer, J., van der Plas, A.: Learning Analytics en Zelfsturend Leren. TNO R10373 (2013)
2. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: evolutionary algorithms made easy. J. Mach. Learn. Res. **13**, 2171–2175 (2012)
3. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
4. Kim, J.: Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Comput. Stat. Data Anal. **53**, 3735–3745 (2009)
5. Kotsiantis, S., Pierrakeas, C., Pintelas, P.: Predicting students' performance in distance learning using machine learning techniques. Appl. Artif. Intell. **18**, 411–426 (2004)
6. Krathwohl, D.R.: A revision of Bloom's taxonomy: an overview. Theory Pract. **41**, 212–218 (2002)
7. Macfadyen, L.P., Dawson, S.: Mining LMS data to develop an early warning system for educators: a proof of concept. Comput. Educ. **54**, 588–599 (2010)
8. Minaei-Bidgoli, B.: Predicting student performance: an application of data mining methods with an educational web-based system. Comput. Educ. **47**, 157–167 (2015)
9. Morris, L.V., Finnegan, C., Wu, S.: Tracking student behavior, persistence, and achievement in online courses. Internet High. Educ. **8**, 221–231 (2005)
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
11. Romero, C., Ventura, S., García, E.: Data mining in course management systems: moodle case study and tutorial. Comput. Educ. **51**, 368–384 (2008)
12. Sánchez-Maroño, N., Alonso-Betanzos, A., Tombilla-Sanromán, M.: Filter methods for feature selection – a comparative study. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 178–187. Springer, Heidelberg (2007). doi:10.1007/978-3-540-77226-2_19
13. Shahiri, A.M., Husain, W.: A review on predicting student's performance using data mining techniques. Procedia Comput. Sci. **72**, 414–422 (2015)

14. Tempelaar, D.T., Rienties, B., Giesbers, B.: In search for the most informative data for feedback generation; Learning Analytics in a data-rich context. Comput. Human Behav. **47**, 157–167 (2015)
15. Wolff, A., Zdrahal, Z., Nikolov, A., Pantucek, M.: Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In: Proceedings of the Third International Conference on LAK'33, pp. 145–149 (2013)