Can machine learning on learner analytics produce a predictive model on student performance?

J. Busch [(1)], P. Hanna, I, O'Neill, A. McGowan & M. Collins

[(1)] Queen's University Belfast, School of EEECS, 18 Malone Road, BT9 6RT
+44 (0)28 9097 4956
j.a.busch@qub.ac.uk

Introduction

The process of learning how to program is wildly regarded as a difficult task [1]. The increasing demand for computing graduates has resulted in increasing class sizes in universities. A positive student learning experience is being diminished due to negative engagement effects with the delivery mechanism [2]. Research into tutor's perceptions of students being highly active during a lecture maybe skewed, assuming cohort engagement levels are higher than they actually are during the term [3]. These factors and many more are directly contributing to low retention rates in computer science and other related disciplines [4].

This study aims to investigate micro-based learner analytics and develop a predictive model to identify students at risk within a large cohort. Using apache web server log data from students undertaking a web development and programming module at a tertiary institute. This data logs student activity on their development and rendering of programming code during the semester which accumulated to over 1.6 million records of data.

Using WEKA [5] workbench to execute machine learning techniques for the analysis of learner analytics we can gain insight into building a predictive model of future student performance. This could eventually lead to a formative monitoring system that can help identify students with low levels of module content engagement irrespective of class size.

This study is founded in learner analytics, which measure, analyse and report on learner's data, so that behavioural patterns and trends can be identified. This in turn can be used to understand and enhance teaching and the environments in which learning occurs [6]. By incorporating data mining and learner analytics this study focuses on a field known as educational data mining (EDM). This is used to predict models rather than explain patterns [7].

Previous research into student web access logs using a quantitative trend-based analysis process suggests a correlation between the student's web server access data and the students' performance. A student's log that shows early access hits at the start of the semester and shows continued hits throughout mid-term of a semester yields a performance significantly better in the assessment when compared with logs of students with little early semester date stamp data [8].

By using a deeper analysis process through a machine learning classifier, the results show a slightly better than 'best guess' prediction rate. The research does allude to the possibility of refining the approach with a bigger data set and a deeper data preparation approach. This study uses a small data set of only 133 instances. The potential of creating a predictive model with higher classification accuracy is evident. The current results are most promising in identifying a low engaged cohort, adding to the research by Ramesh et al [9], and allowing an automatic early intervention flagging system to be developed so students as risk can be identified.

Educational Data Mining

The study uses binary classification as the applied data mining approach, to employ a set of pre-labelled instances to create a predictive model that can classify a given dataset. The classification framework used can employ decision tree (DT), random forest (RF), and support vector machines (SVM) classification

algorithms. By reporting on each models training and testing classifier accuracy rating, from a singular dataset a cross comparison evaluation can be deployed.

Related research work in this area in identifying the potential of using machine learning supervised classification algorithms on learner analytic data to predict student performance can return high correction rates. Ramesh et al [9] reported that a Multi-Layer Perception (MLP) classifier gave a 72.38% accuracy rate for predicting student performance. Baradwaj [10], used a DT model to create classification rules to detect those students that might require learning support. Kumar [11] compared several DT algorithms and evaluated that multivariate regression prediction MP5 model produced the most accurate rate of 97.17%. Bhullar [12] reported a 77.14% success rate using a C4.5 DT algorithm to predict student at risk of failing. These case studies primarily use summative profile data, e.g. past assessment grades and engagement metrics. Other external demographical data, e.g. sports activity, is also used to build the dataset. Our research focuses not on the student macro learner analytics, but the micro-learner analytics which are gathered from an apache web server used for software development.

Research in the area of programming and educational data mining used an FP-Tree model to find frequent patterns in their large dataset. The identified frequent patterns were used as variables to run a k-means clustering model classifier algorithm which concluded that students living in an urban location show a higher programming skill set their rural counterpart [13].

Data mining learner analytics from a programming assessment was carried out by Blikstein [14]. The study attempted to uncover student programming behaviours using custom software to log a student's activity. The analysis did not attempt to use machine learning, but rather solidify the assertion that analysis of logging programming activity data will show definite patterns. Following on from the study, Blikstein et al [15] analysed a bigger dataset using machine learning supervised regression and unsupervised x-means clustering techniques. The results showed that the programming learner data showed some well-defined clusters but it did not correlate clearly to performance.

Micro-based Learner Analytics

Data acquisition was performed using data collected during the 2014-2015 academic year from an undergraduate cohort undertaking a web development and programming module delivery at a UK based University. The access log files from the period between October and December 2014, (73 days), held on the Apache web server which stores HTTP requests for each individual were stored. Students are required to develop PHP code and run it through this Apache web server to complete a large individual-based project/course work. It can therefore be reasonably assumed the log data records the student's engagement with the modules learning material and assessment task.

```
143.117.101.245 - - [27/Oct/2014:13:33:10 +0000] "GET
/lab04a/ HTTP/1.1" 404 301 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:32.0) Gecko/20100101 Firefox/32.0" 13428
```

Figure 1 – access log instance.

Figure 1 shows the data for a typical logged instance. The access log data provides a deeper or micro-level learner analytic data because of its detail. Every HTTP request to the web server is collected and logged. Details on date, time and request type are gathered. A derived variable can be created from the logged details which can indicate how long the students was active.

The merit of using access log files is that they are a standard non-proprietary data format that does not need additional software to record and store the data. All web server software, by default, generates and stores similar access log data. By focusing on data that is automatically generated removes the need to rely on external software solution to gather and store learner data.

Objectives

The main objectives of this research are

- Identification of influencing predictive attributes from apache log data set
- Analysing the classification algorithms on the given data set.

Methodology

As illustrated in Figure 2, this work utilised a sequential process to collect, prepare and ultimately classify a suitable dataset.

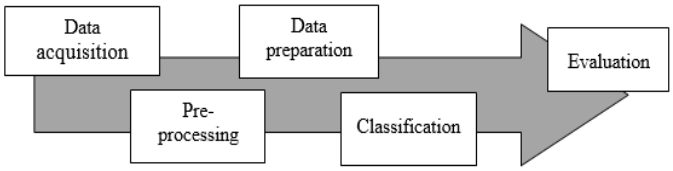

Figure 2. Machine learning process to produce a predictive model

The access logs provided the raw data of over 1.6 million logged instances. Pre-processing was applied using a custom tool to filter the data. This filtering process cycled through the log files of 133 student access log files and generated a dataset of 133 instances. The attributes are shown in Table 1, with the majority being derived using the date and time data held within each log instance.

Table 1 - Pre-processed student related variables for supervised learning

Attribute	Description	Type
access_hits_total	Total amount of access hits within the 73 days	Non-derived {numeric}
early_dur_secs_total	Total amount of seconds calculated on first 23 days.	Derived {numeric}
early_hits_total	Total amount of access hits within the first 23 days	Non-derived {numeric}
early_engaged_average	Calculated Total Activate Days / Total Days (23)	Derived {numeric}
mid_dur_secs_total	Total amount of seconds calculated on middle 25 days.	Derived {numeric}
mid_hits_total	Total amount of access hits within the middle 25 days	Non-derived {numeric}
mid_engaged_average	Calculated Total Activate Days / Total Days (25)	Derived {numeric}
late_dur_secs_total	Total amount of seconds calculated on last 25 days.	Derived {numeric}
late_hits_total	Total amount of access hits within the last 25 days	Non-derived {numeric}
late_engaged_average	Calculated Total Activate Days / Total Days (25)	Derived {numeric}
grade_flag (class)	Class attribute, profile achieve > 50% assessment grade	Derived {binary 0,1}

This study focuses on grade prediction using single binary classification to identify students at risk. This is because the available dataset was unbalanced and prone to overfitting in a multi-class classifier scenario aiming to identify students falling in the traditional grade boundaries. i.e. 43.6% of the instances where within the 60%-69% grade classification (2.1), whilst there were much smaller proportions in the other boundaries (70%+, 50-59%, 40-49% and 0-39). The dataset had a natural split with 54.1% of the instances above a 50% mark awarded and 45.9% achieving below the 50% mark so a binary classification task was focused on for the purposes of this experiment.

The *late_* data, relating to student activity in the last 25 days of the assessment period is logged and explored. However if a truly predictive model is to be realised which can be used to identify at risk students during course delivery, it is envisaged that these data points would ultimately be excluded.

The final stage of pre-processing is to apply. In line with standard procedures, numerical features were normalised and instances randomised. By randomly ordering and normalising the dataset the numerical data is made relative so as to avoid different scales which can skew the machine learning classification process [15].

Data preparation involves feature selection [16], to create a dataset for building a good predictor [17]. A wrapper attribute evaluator methodology addressed the variable selection process. WEKA modelled the ranking data of the dataset's attributes, Table 1. It involves searching through all possible combinations of attributes in the dataset to find which subset of attributes will be best for the final feature set for predicting the class attribute. The decision tree C4.5 (DT), random forest (RF) and support vector machines (SVM) classifiers were selected to determine the method and to assign a weight to each subset

of attributes. A greedy stepwise algorithm was then needed to perform the search. Table 2 shows the top 5 ranked attributes.

Table 2 – Top ranking attributes using a wrapper evaluator

Classifier	Average Rank	Swing -/=	Attribute
DT	3.1	1.81	early_hits_total
DT	3.7	1.62	mid_hits_total
DT	4.5	2.16	early_dur_secs_total
DT	407	2.65	early_engaged_average
DT	407	2.97	late_engaged_average
RF	109	2.12	early_engaged_average
RF	2.4	1.11	late_engaged_average
RF	4.1	2.12	early_hits_total
RF	4.9	1.97	late_dur_secs_total
RF	6.5	1.28	mid_dur_secs_total
SVM	1.6	1.8	late_engaged_average
SVM	3	1.26	early_dur_secs_total
SVM	3.9	2.02	late_dur_secs_total
SVM	4.9	0.94	access_hits_total
SVM	4.9	2.51	early_hits_total

Results and Discussion

The top ranking results show a positive trend with the *early_* attributes and *mid_* attributes accounting for 60% within all classifiers. This might suggest that the *late_* attributes can be discounted within the feature set as the classification model does not rank the attributes as good discriminators in the prediction analysis. The SVM highly ranks the *late_* attributes which suggests it would make it an unsuitable algorithm for using a predictive model for our dataset. Whereas the DT classifier is showing a more positive preference to the *early_* and *mid_* attributes, therefore it could return a higher prediction rate on these attributes.

The dataset was tested with three different classification algorithms: decision tree C4.5 (DT), random forest (RF) and support vector machines (SVM). Running a 10-folds cross-validation on each classifier generates a separate dataset so that enhanced data is generated on the train set. Classifiers are built until finally, the evaluations were applied to the original test data. The final results were collected from average of ten run-times. This produces a confusion matrix that creates sensitivity and specificity measurements for analysis to evaluate the performance and either support or reject the hypotheses.

A summary of the sensitivity measurements (true-positive detection rate) and specificity (true-negatives detection rate) of each classifiers' performance on the dataset is shown in the Table 3. Each test is also showing the number attributes used. Figure 3 identifies the attributes used in each classification process.

Table 3 – Summary of measurements

Classifier	Measurement	Attri_11	Attri_7	Attri_6
SVM	Accuracy rate	67.7%	64.7%	54.8%
SVM	Sensitivity (TP)	0.68	0.67	0.56
SVM	Specificity (TN)	0.67	0.62	0.51
DT	Accuracy rate	63.1%	60.9%	60.1%
DT	Sensitivity (TP)	0.63	0.6	0.59
DT	Specificity (TN)	0.64	0.62	0.61
RF	Accuracy rate	60.9%	58.6%	57.9%
RF	Sensitivity (TP)	0.62	0.61	0.60
RF	Specificity (TN)	0.58	0.55	0.61

The results show that data mining algorithms classifiers have performed similarly and are below a good standard of predication. The accuracy rate of the SVM model using 11 attributes results in a 67.7%. Although the specificity measurements show that the model has great difficulty correctly predicting a student who is not meeting the condition of interest. Figure 4 shows the confusion matrix values for the SVM 11 attributes experiment, the 'a' label refers to students who are above the 50% threshold and the 'b' label refers to students not above the 50% threshold. The ground truth labelling of the instances in the dataset include 61 students considered in the 'at risk' category. But the model is returning a total of 35 (as highlighted in figure 4). Therefore correctly classifying only 57.3% of the dataset. This is far from perfect, and is not much better than a 'best guess'.

a	b	
55	17	a = 1
26	**35**	b = 0

Figure 4 – SVM 11 Atttribute Confusion matrix

The SVM model using 11 attributes is better at predicting students that are above of 50% threshold. There should be 72 students being returned, but the

test produces 55, which returns a more positive successful prediction rate of 77.5%. Since the predictive model is trying to identify students not meeting a certain condition the success rate is relevant but not what the researchers had hoped for.

The other 8 tests show little difference in their ability to correctly detect an instance not meeting the condition. The most successful model that determines the greatest number of those instances that are below the threshold is the SVM experiment using the 7 attributes with an accuracy rating of 64.6%. It correctly identified 36 instances as being below the threshold which equates to a success rate of 59.0%.

Attributes	**Attri key**
access_hits_total	11
early_dur_secs_total	11, 7, 6
early_hits_total	11, 7, 6
early_engaged_average	11, 7, 6
mid_dur_secs_total	11, 7, 6
mid_hits_total	11, 7, 6
mid_engaged_average	11, 7
late_dur_secs_total	11
late_hits_total	11
late_engaged_average	11
grade_flag (class)	11, 7, 6

Figure 3 – Definitions of attributes name and key

This is far from ideal, but the results do suggest that the models can perform better with only the *early_* to *mid_* attributes. This shows that predicting the student performance is possible using early to mid-semester learner analytics.

The results are disappointing and seem to suggest that the classifier models have great difficulty in training from the given dataset alluding to poor data quality. The instances do not have enough discriminatory power on either side of the threshold. The authors suspect that there are too many instances that are close to the boundary on either side of the 50% threshold. Further work in building a more discriminating dataset is required.

The identification of influencing predictive attributes from apache log data has been a success and the study's results that *early_* and *mid_* variables can be related on as part of the feature set. The data mining algorithms show potential but their poor prediction rate on instances at risk are not sound enough.

Conclusions

Several studies have used EDM and more specifically machine learning classification algorithms to improve the quality of education, identifying students failing to engage and struggling to cope with the material.

In this research we addressed the use of apache web server data to predict if a student's performance was above or below a fixed threshold. The results show potential in future study and research, machine learning techniques can use early to mid-semester micro-learner analytic data to classify a prediction.

More work needs carried out in three areas. Firstly, an increase in the instance population would be required, this will balance that instance data by reducing the effect of overfitting. It is also envisioned that the increased dataset will also help the classifier return a lower sensitivity with higher specificity measurements. The second area of exploration is to expand the data points examined through the addition of the error logs that are also produced in tandem with the access logs. This would introduce extra variables that might improve the feature selection results and in turn improve the prediction rate. Adding more general demographic data, e.g. past academic performance, to the dataset needs to be explored.

If future initial investigations into a binary classification model proves successful, with expanded datasets, it may also be possible to explore multi-class classification at a more granular level.

References

[1] Milne, I. & Rowe G. *Difficulties in Learning and Teaching Programming - Views of Students and Tutors*. Education and Information Technologies, (7), (2002) p. 55-66.

[2] Cuseo, J. *The empirical case against large class size: adverse effects on the teaching, learning, and retention of first-year students*. The Journal of Faculty Development, (21), (2007) p. 5-21.

[3] Fassinger, P., A. *Professors' and students' perceptions of why students participate in class.* Teaching Sociology (1996) p. 28.

[4] Giannakos, M. N., Pappas, I. O., Jaccheri, L. & Sampson, D. G. *Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness*. Education and Information Technologies, (2016) p. 1-18.

[5] Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/

[6] Phillips, R., Maor, D, Cumming-Potvin., Roberts, P., Herrington, J., Preston, G. & Moore, E. *Learning analytics and study behaviour: A pilot study.* In G. Williams, P. Statham, N. Brown & B. Cleland (Eds.), Changing Demands, Changing Directions. Proceedings ascilite, (2011) p. 997-1007.

[7] Atif, A., Richards, D., Bilgin, A. & Marrone, M. *Learning analytics in higher education: a summary of tools and approaches*. In 30th Australasian Society for Computers in Learning in Tertiary Education Conference, (2013) p. 68.

[8] Busch, J., Anderson, N., McGowan, A., Hanna, P. & Collins, M. *Using Temporal Learner Analytic Data to Develop a Casual Predictive Model.* European Conference on Educational Research, ECER (2016).

[9] Ramesh V.A., Parkavi P. & Ramar K. *Predicting student performance: a statistical and data mining approach*. International journal of computer applications, (8), (2013) p. 35-39.

[10] Baradwaj BK, Pal S. *Mining educational data to analyze students' performance*. International Journal of Advanced Computer Science and Applications, (6), (2011) p. 63 -69.

[11] Kumar S.C., Chowdary E.,D., Venkatramaphanikumar S. & Kishore K.,K. *M5P model tree in predicting student performance: A case study.* In Recent Trends in Electronics, Information & Communication Technology (RTEICT), (2016) p. 1103-1107.

[12] Bhullar M., S. & Kaur A., *Use of data mining in education sector.* In Proceedings of the World Congress on Engineering and Computer Science, (1), (2012) p. 24.

[13] Arockiam L., Charles S., Carol I., Thiyagaraj P.B., Yosuva S. & Arulkumar V. *Deriving Association between Urban and Rural Students Programming Skills*. International Journal on Computer Science and Engineering, (3) (2010) p. 687 – 690.

[14] Blikstein, P. *Using learning analytics to assess students' behavior in open-ended programming tasks.* In Proceedings of the 1st international conference on learning analytics and knowledge (2011) p.110-116.

[15] Blikstein P., Worsley M., Piech C., Sahami M., Cooper S, & Koller D. *Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming.* Journal of the Learning Sciences, (4), (2014) p. 561-599.

[16] Guyon I, Elisseeff A. *An introduction to feature extraction,* Feature extraction, Springer Berlin Heidelberg, (2006) p. 1-25.

[17] Guyon I. & Elisseeff A. *An introduction to variable and feature selection*. Journal of Machine Learning Research, (3), (2003) p. 1157-1182.