Using Educational Analytics to Improve Test Performance

Owen Corrigan[1], Alan F. Smeaton[1], Mark Glynn[2], and Sinéad Smyth[3]

[1]Insight Centre for Data Analytics
[2]Teaching Enhancement Unit
[3]School of Nursing and Human Sciences
Dublin City University
Glasnevin, Dublin 9, Ireland
Alan.Smeaton@DCU.ie

Abstract. Learning analytics are being used in many educational applications in order to help students and Faculty. In our work we use *predictive* analytics, using student behaviour to predict the likely performance of end of semester final grades with a system we call PredictED. The main contribution of our approach is that our intervention automatically emailed students on a regular basis, with our prediction for the outcome of their exam performance. We targeted first year, first semester University students who often struggle with making the transition into University life where they are given much more responsibility for things like attending class, completing assignments, etc. The form of student behaviour that we used is students' levels and types of engagement with the University's Virtual Learning Environment (VLE), Moodle. We mined the Moodle access log files for a range of parameters based on temporal as well as content access, and use machine learning techniques to predict likely pass/fail, on a weekly basis throughout the semester using logs and outcomes from previous years as training material. We chose ten first-year modules with reasonably high failure rates, large enrolments and stability of module content across the years to implement an early warning system on. From these modules 1,558 students were registered for one of these modules. They were offered the chance to opt into receiving weekly email alerts warning them about their likely outcome. Of these 75% or 1,181 students opted into this service. Pre-intervention there were no differences between participants and non-participants on a number of measures related to previous academic record. However, post-intervention the first-attempt final grade performance yielded nearly 3% improvement (58.4% to 61.2%) on average for those who opted in. This tells us that providing weekly guidance and personalised feedback to vulnerable first year students, automatically generated from monitoring of their online behaviour, has a significant positive effect on their exam performance.

Keywords: learning analytics, mining educational data, predictive analytics, machine learning

1 Introduction

Automated, weekly, personalised emails to students is an excellent adjunct to support from the Lecturer, tutors, lab supervisors or other student supports, but even with these in place, it is inevitable that some students will *slip through the cracks* because of the sheer numbers of students on some modules. As a motivating example, consider a lecturer who teaches a first year engineering module with +200 students. The lecturer may not have the time to deal with individual students, or be able to identify who needs help most urgently. If we automatically identify which students are struggling based on the characteristics of their online behaviour, then we could contact them directly and provide them with the resources they need to succeed. If we provide continuous feedback then they can monitor their own progress. The earlier we can intervene with students, the better.

In our University, students take written exams after taking a module, which is a 12-week course in a subject (e.g. Introduction to Law). Each module that we work with is at least partially graded based on a final end-of-semester written examination percentage mark.

We took data generated from students' online behaviour, and useed it to improve their learning experience and specifically, their performance in end-of-semester written examinations. We used log data from the University's online virtual learning environment, Moodle. We then combined this with past exam performance to build a predictor which accurately classifies whether a student in the current cohort of students is likely to pass or fail the module. Section 3 describes how this classifier was built.

This classifier leverages online behaviour and examination outcomes from past students, in order to inform current students as to how they are progressing. We target University students in their first semester when they are most vulnerable and often feel lost or overwhelmed by what is for most, a sudden change to University life. We use past, and present, log data to predict likely outcomes on a weekly basis and naturally the accuracy of our predictions is likely to get more accurate as the module progresses. As a form of alerting, students receive emails each week that advises them to study more, that they really need to study more, or that they seem to be doing OK, whatever is appropriate. Section 4 describes how the details of this intervention worked in practice.

For nine of the ten modules for which we ran the alerting service, examination consisted of a composite of continuous assessment and an end of semester exam. One subject was entirely exam based. Section 5 describes how we evaluated this experimentally and the results that we obtained.

The next section lists some related work in this area and describes how this related work compares to ours.

2 Related Work

There are several examples of using students' online behaviour to either monitor student performance or predict their performance in exams. The Open Univer-

sity ran a project to identify "at-risk" students in order to improve retention [9] and this complements our work in several ways. Similarly to our project, they used predictive modelling with inputs being engagement with VLE resources, assessment performance, previous exam performance and demographics of the students. However in their study they found that demographics play a less important role than VLE engagement. They also found that the best approach is to analyse data on a module-by-module fashion instead of across all modules, the same approach as we have taken.

The size of the Open University study is impressive, with 7,000 students targeted across three modules, whereas our study targeted 1,200 students across 11 modules. Their prediction target was similar to ours as they predicted failures in a module at regular intervals during semester, specifically, every time there was an exam. In contrast, our interval was weekly and our approach differs from theirs in several ways. They deal with a much wider variety of students, many working part full-time while doing modules, whereas ours were a more homogeneous set of students. However the main difference between approaches is that the Open University work predicted the set of students most in need of help, and left it up to the module organiser to intervene. In our work we alerted all students on the module directly, to inform them of their progress and likely outcome.

An influential early paper by Romero, Ventura and Garcia [8] describes a four-step framework for analysing VLE data. These steps are: collect data, process data, perform data mining/machine learning steps and deploy results. It also describes how data mining and machine learning methods can be applied to the data. That paper inspired the Purdue Signals project [1] which takes a variety of factors such as points on the module to date (which are assigned by the module instructor), time spent on a particular task and past performance in other exams. It then uses this to predict what a students score will be. The Purdue Signals project uses a different mechanism to deliver predictions compared to our work, although both are targeting the student. In the Purdue Signals system, students can see how well they are doing via a web application showing if they are classified as "green", "orange" or "red" for a given module. Another differentiating aspect is that our work uses the VLE data in a different way, updating the predictions automatically every week.

In a paper by Calvo-Flores et. al. [2], the authors also predicted user exam performance based on VLE log data. Their aim was to find students who are in need of learning reinforcement. This work established that features derived from the access logs were enough to predict success with a high degree of confidence. Some of the features that they use are ratio of resources viewed and total resource views. In their case they tested it on a module with over 240 students and had a prediction accuracy of above 80%. Our paper uses some of these same features, but we retrain our model for each week, allowing us to recalculate predictions on a weekly basis, which make our insights more actionable.

In [3] the authors performed a statistical analysis on the access logs of a VLE for three modules. They examined many features extracted from these access

logs, checking if they were correlated with the final exam mark the students achieved. These features included VLE activity, unique page views, number of log ins, whether they were on or off campus, the coverage of resources the students accessed, and the effect of accessing Moodle on different days of the week. This work was useful in establishing which features would be useful to extract from the logs.

Similarly, in [4] work by Cocea and Weibelzah on feature extraction was an influence on ours. The features used in this study include number of pages accessed, time spent reading pages and student performance on mid-semester tests. Even though they were using a VLE targeted at one particular module, the features used were similar to ours. However their objective was different, with their goal being to estimate learners' levels of motivations.

In the next section we describe the raw data we used, how we processed it to extract features and train a series of machine learning classifiers, and then applied it to live student behaviour logs in order to alert students directly.

3 Data Analysis

3.1 Periodicity of VLE Logs

The basis for student behaviour that we used to feed our performance predictions was students' online interaction with the VLE, Moodle. By default, Moodle records every instance of a student accessing a Moodle page of any kind and records the page, date and time, the student identifier, and the IP address of the device used to access. In order to predict performance we need to be able to train on past student performance where the outcome is known and hence we need to work with modules where every year the log files follow the same patterns of access. Each module, by its very nature, will have different access frequency patterns which are influenced by scheduling of lectures, lab sessions, group sessions, mid-semester tests, assessment deadlines and final exams. Plotting the overall student activity for each module by simply counting the number of student accesses allows us to determine visually if the module displays an annual periodicity. This is useful for us in determining the set of modules for which there is enough training material from past years which can be used. For instance, if a module drastically changed in content or in delivery within the last year or two because a new lecturer took over and removed class tests, this would show up in the activity levels and would mean that because the training material has changed, that module would not be suitable as a basis for making weekly predictions. Other changes which do not affect scheduling of the module, such as adding additional content or modifying an exam, would not show up in these access or usage as much. We will compensate for this later by building a classifier and cross-validating the results of the classifier's prediction accuracy. If the prediction accuracy results are poor, perhaps because the course has changed so much that our models are not useful, then we would not use that model.

Figures 1 and 2 show the aggregate activity levels for all students over the past five years and each demonstrates a regular annual periodicity meaning that

student accesses from previous years, coupled with the performance of those students in those previous years, provides suitable training material for classifying this year's cohort of students.

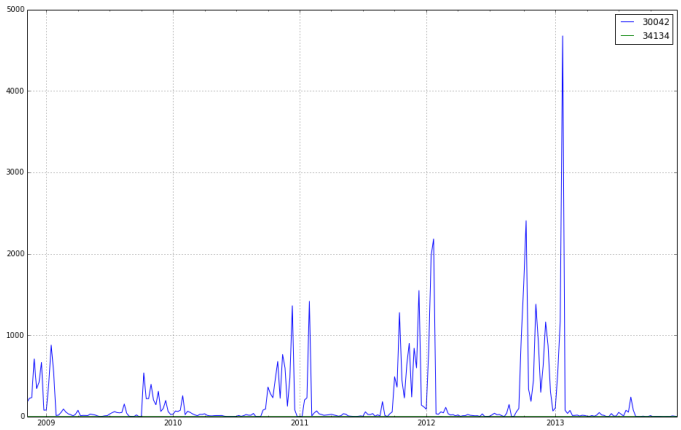


Fig. 1: Activity levels for module CA103 (Computer Systems Hardware)

3.2 Data Preparation

We initially receive Moodle logs in a format which contains the user id, module, timestamp, IP address and the Moodle resource being accessed. To this we add the result that each student obtained in the written end-of-semester exam, which constitutes the largest part of how the overall grade for the module is computed. However we then have to transform this raw data by extracting a set of features associated with each student and module which, in particular, focuses on temporal information associated with the logs.

3.3 Feature Extraction

We processed the Moodle logs by extracting a variety of features for each students' access to each module, in each year. One of the benefits of the support vector machine (SVM) [7] we used to build the classifiers, as described later, is that not all of these features have to be actually useful in terms of discriminating different forms of student outcome for the module. That means we can be open-minded about how we represent students' online behaviour and if a feature is not discriminative, the SVM learns this from the training material, i.e. from student data from previous years. We extracted the following three features from the logs

– A simple count of how many logs each student accessed.

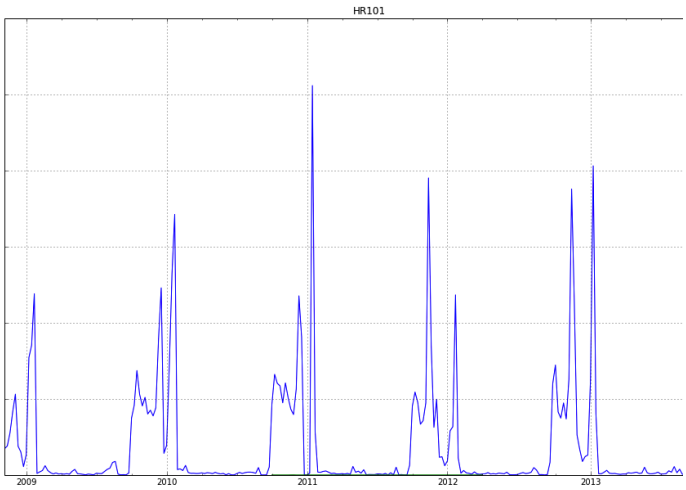


Fig. 2: Activity levels for module HR101 (Psychology in Organisations)

- The average time that the logs were accessed.
- The ratio of clicks on weekends vs. weekdays.
- The ratio of on campus to off campus accesses. This was determined using the IP address.

We needed to take into account the fact that we would be downloading the logs as a constant stream. We updated the predictions once a week with the new data obtained. When training we added an additional column to the log data, which was the academic week number that the log took place. We then divided the log data into 12 separate columns, depending on which week it took place in and filtering out any accesses which took place after week 12. In the next section we will discuss how this was used in classification.

Corresponding to each student and set of features, we extract a student result. This was a simple binary variable, either a pass or a fail.

3.4 Classification

Once the features were extracted for past deliveries of modules, we built a set of classifiers, one for each week of semester and right up to the week of the final examination, which predict the student's likely pass/fail in the end-of-semester final grade. In most of the University's modules the examination pass rate is between 70% and 95% and so an issue with the data is that because passes are a lot more common than fails we need to re-sample the data so that the prediction result is not biased towards the most common case. Another issue is that there can be a lot of features and not a huge amount of students and so we have to be wary of over-fitting and choose modules which have large numbers of students.

To determine the accuracy of our predictions we used the Receiver Operator Characteristic Area Under Curve (ROC AUC) relevance measure [5]. We used this because it is more resilient to imbalanced classes than other measures, such as simple accuracy. An ROC AUC score greater than 0.5 is better than random. In evaluating the usefulness of different feature combinations we use cross-validation and because the number of samples can be small we use 5-fold validation which reduces the variability of the obtained prediction accuracy score.

Using the SciKit-Learn library [6] an SVM for classification was created. It was initialized with the following settings.

- A 'C' value of 0.1 was found to work well;
- A linear kernal was chosen;
- Probability estimates were enabled so that the ROC score could be generated. These also allowed us to rank the students in order of our predictions as to their likelihood of pass/fail for the module;
- The class weights were set to be inversely proportional to the class frequencies. This helped address the over-fitting problem.

Training the classifier involves

- Fitting the classifier with 80% of the historic log data. The features are the from the features listed above. The target variable is whether the student passed the module that year.
- Testing the remaining 20% of the historic log data and generating a prediction and a confidence score.
- Comparing these confidence scores with the known target variable to calculate the AUC ROC score
- Repeating this process with a different 20% test set and taking the average to give a more accurate score.

We then trained 12 separate classifiers, one for each week of semester. This involved repeating the above proceedure 12 times. The classifier for the first week only contained the first week of data. The second including the first two weeks of data (to represent the situation in the second week), and so on. Each week 4 new features are added to the total number of features, so that by the end of week 12 we were making predictions based on 48 features in total.

To determine how effective a the set of 12 classifiers are over the semester, and to gauge at which point during the semester it is reliable enough for us to begin sending alerts, we calculate this measure for classifiers built for each of the 12 weeks of access log data that we have extracted features for. The features are then scaled using the standard score, and the transformation is saved so that it can be applied to the live data later.

As a rough heuristic, if the resulting ROC AUC value is at or below 0.5, we judge that the classifier is not performing better than random. Once this line is above 0.5 consistently, we regard the classifier as working. In our case, the best modules typically have a value between 0.6 to 0.7 for the weeks following week 3

or 4 of our 12-week teaching semester. This helps us to determine what week to start sending the predictions to students, and also to compare the performance of different classification algorithms on the same data. For examples, see Figures 3 and 4. For the SS103 module, we decided that the ROC AUC score was consistently above 0.5 from week 3 onwards, so we started the email interventions then. For the MS136 module, the predictions were better earlier on, and so we started sending thise predictions to students in week 2.

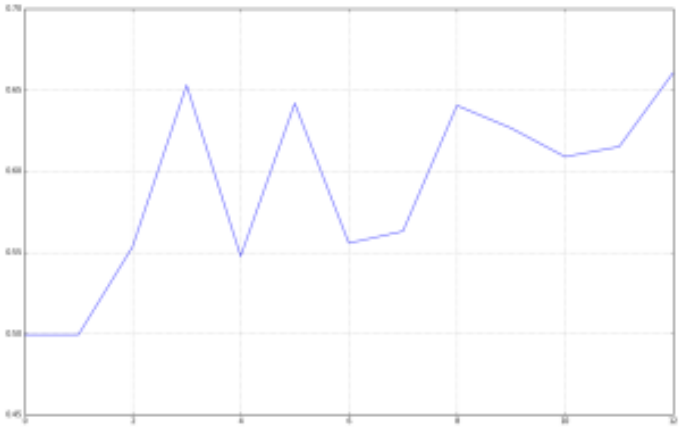


Fig. 3: Prediction accuracy for SS103 (Physiology for Health Sciences) by week

3.5 Selecting Modules

We ran this procedure of assessing the suitability of modules across all of the larger modules given to first year students in Semester one in the university and then filtered them based on a number of criteria. The first is that there must be a minimum amount of students in a module, which we chose, to be 80 in order to have enough student instances so as to make the classification worthwhile. We also wanted to have the weekly alerts delivered to the largest number of students. Secondly, since the method of predicting pass/fail will work poorly if the module has a high pass rate and thus a classification bias, we chose modules which had a pass rate of less than 90% for the examination component. Thirdly, we could only work with modules where the Moodle logs display periodicity as described in section 3.1 above. Finally, when we plot the ROC AUC values for those modules remaining, only those which have a consistent value above random for a number of weeks were kept for the study. This left us with 10 modules outlined in Table 1 and a total of 1,558 eligible student registrations once repeat students, those under 18 years of age and visiting students from other Universities were eliminated. One limitation of this process is that we

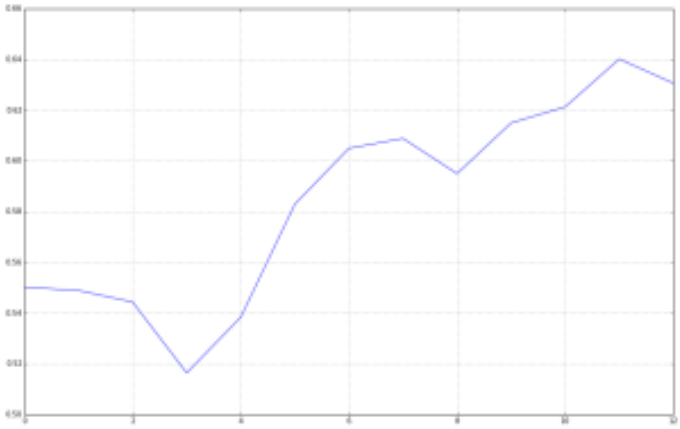


Fig. 4: Prediction accuracy for MS136 by week (Mathematics for Economics and Business)

cannot run this on a new module, as we require data for at least a year to create our models.

To summarise, processing the data takes place in three steps as shown on the diagram below in Figure 5. The first involves simply getting the activity levels of the modules and filtering modules them down based on criteria discussed earlier. The second step involves extracting features and producing classifiers on previous data. The final stage involves making predictions on current log data, and emailing the results to students as an intervention.

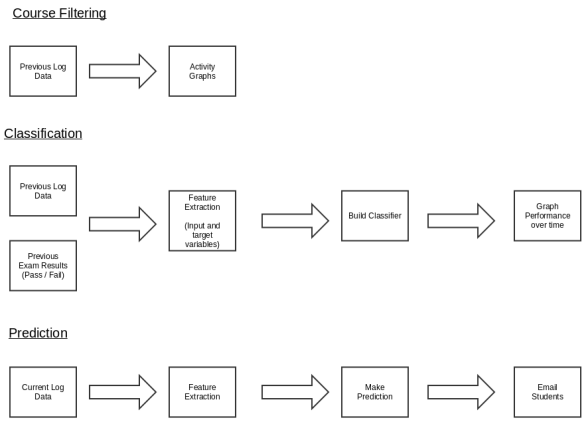


Fig. 5: Summary of data processing steps

4 Student Interventions

For each of the 10 modules in Table 1 and for each week after our predictions reach a usable accuracy, our classifier outputs a pass/fail prediction for each student taking the module, plus a confidence value for each. We used the output from this to rank students from least likely to fail to most likely to fail and we also have a tipping point or threshold pointing to the likely overall pass rate for the module. We now examine how we can use this information to feed back directly to students and to do this we tested two separate strategies.

For students in some of the modules we sent weekly emails based on our prediction of whether they would pass or fail. For each of these two groups we divided each into two more groups resulting in four groups — "bad" and "poor" for those predicted to fail, "good", and "great" for those predicted to pass. The rationale behind this is that the best way to divide the groups may be based what we think will happen with their results. Each group is sent a weekly email with bad for instance receiving an email saying how students need to work harder, while those in the great group told they are working well on the module. Each email contains pointers to resources the student can use, such as contact details for student support services and for their lecturer.

For students on each of the other set of modules, they were broken into 10 equally sized groups based on the overall prediction ranking and the emails informed each student of what percentile group among the class they fall into. The bottom 50% of students are given a more encouraging email than the top 50%, the idea being that students get an idea of how they are doing relative to their peers in the class. This might encourage some competition among students, incentivising them to and hopefully they would see the results of working harder sooner in the feedback loop.

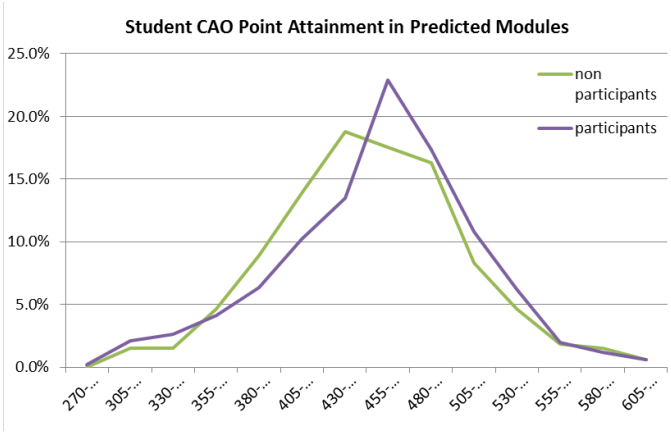
Each week this process is repeated with the most recent log data downloaded from the Moodle log files. Predictions are generated for each module in the manner described above. For each week of semester, each student was sent a new email. All this was done with approval from the University's Research Ethics Committee, the University's Data Protection Officer, student services support, and the Module Co-ordinator's of the 10 modules. In addition, students were presented with a plain language statement of what their data was being used for and presented them with the choice to opt-in or opt-out of receiving the weekly alerts.

In the next section we will analyse the impact that receiving weekly prediction alerts had on student performance.

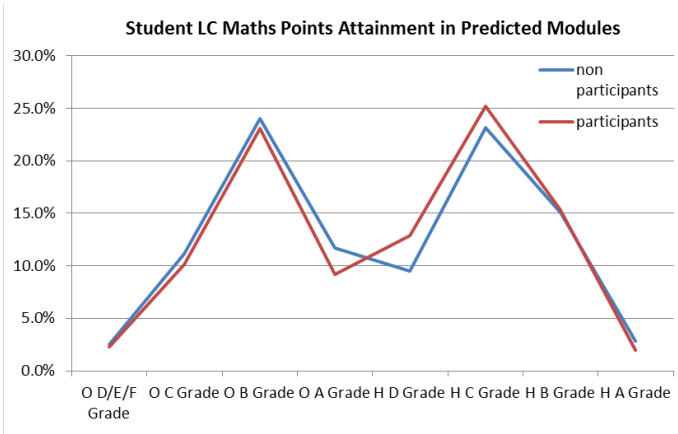
5 Results

We first examined the relationship between the scores that students obtained in their Leaving Certificate exams, equivalent to SAT tests in the United States, and whether they opted-into or out of the alerting service. For both conditions we plot both the University admission points (known as CAO points) that the

student obtained in the Leaving Certificate, which is the sum of a students top six subjects (maximum 625). We also plot their grades in the exam in mathematics as that subject is regarded as a strong indicator of academic ability. We can see from Figure 6 that there is little difference between those who opted-in to the alerting and those who opted out. This indicates no difference in the entry profiles of participants vs. non-participants overall and therefore there is no spurious relationship between the academic ability of the student and whether they opted-in to PredictED or not. This allowed us to gauge the impact the system had on students. However this opt-in method may have other issues, such as motivation differences, compared to a truly randomised group.



(a) CAO Points at University entry



(b) LC Mathematics at University entry activity

Fig. 6: CAO points and Leaving Certificate Mathematics results for student participants and non-participants

We also examined the relationship between the predictions that we made, and the actual results that the students ended up achieving overall. We did this by generating a binary pass or fail prediction for each student every week throughout the semester. Once the exams were over and the results available, we compared the predictions against the actual results that students achieved. We then counted the number of true positives, false positives, true negatives and false negatives and calculated the F1-score from this matrix for each week. The F1-score was used due to the highly imbalanced classes present in this results matrix and the F1-score results can be seen in Figure 7. We can see from this figure that the F1-score remains consistently high, especially over the latter weeks of the semester.

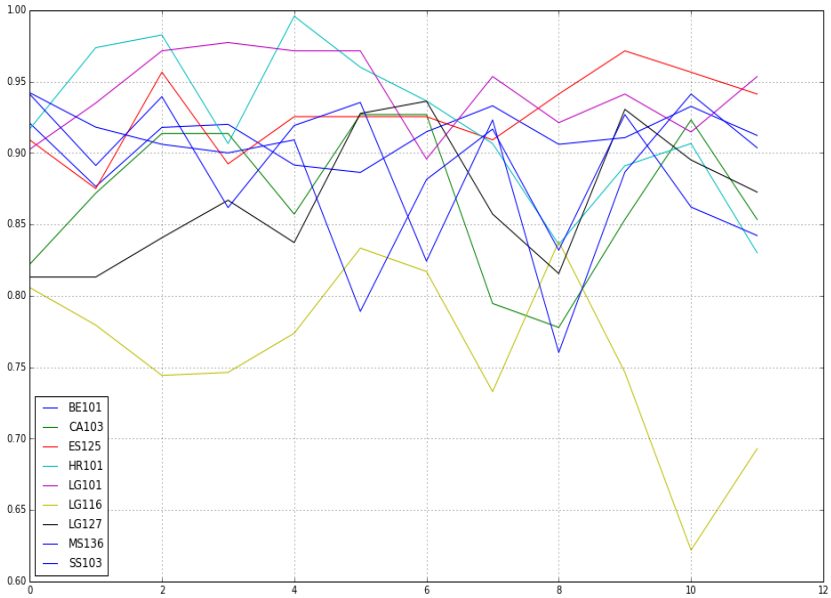


Fig. 7: F1 scores by week. Each coloured line represents a different module

The most important result that we obtained was when we looked at the difference between those who opted-in to receiving emails each week, compared to those who didn't opted-out. Table 1 shows this comparison and the last row indicates that for those who opt-in they can expect to see an average increase of +2.67% in their actual exam marks, all other things being equal. The final column of Table 1 shows the impact of PredictED on a per-module basis with ✓✓indicating significantly higher performance for participants (students) and ✓indicating higher performance. This table shows that for 3 modules, BE101, HR101 and MS136, there was a significant improvement in students' exam performance while for 5 other modules there was improved performance. Only 47

students (of a possible 67) registered to take part in PredictED as part of ES125. This low number of registrations for the module as a whole was unexpected but we decided to continue and make the weekly alerts available nonetheless. The average decrease of 0.5% in exam performance can be discounted based on the low N for this group. One area for future research would be on what the impact of the text of the emails had on the students.

Table 1: Performance in end-of-module examination showing number of eligible students for module (Regs), and comparing average non-participants' exam mark (Non) vs. participants' average exam mark (Part.) and significance of difference between these (Signif.)

Code	**Module Title**	**Regs**	**Non**	**Part.**	**Signif.**
BE101	Intro. to Cell Biology & Biochemistry	284	58.9%	62.1%	✓✓
CA103	Computer Systems Hardware	122	70.3%	71.3%	✓
CA168	Digital World	78	63.8%	65.3%	✓
ES125	Social and Personal Development with Communication Skills	47	67.0%	66.5%	-
HR101	Psychology in Organisations	152	59.4%	63.3%	✓✓
LG101	Introduction to Law	172	53.3%	54.9%	✓
LG116	Introduction to Politics	154	45.7%	44.9%	-
LG127	Business Law	288	60.6%	61.8%	✓
MS136	Mathematics for Economics & Business	157	60.8%	69.4%	✓✓
SS103	Physiology for Health Sciences	104	55.3%	57.0%	✓
Across all modules (weighted by participation)			**58.58%**	**61.25%**	✓✓

6 Conclusions

In this paper we described a method of predicting students outcomes in a first year University module or subject based on feature extraction from VLE access patterns, and the impact of feeding these predictions directly back to students on a weekly basis during semester. Our mechanism to deliver this information was to email students directly with a tailored message based on their predicted outcome for the module. A new message was sent every week to approximately 1,200 students who opted into this service, so that they could see how they are performing relative to the rest of their class. We developed a new method for constructing these predictions every week. We also demonstrated a method for evaluating which modules were best suited to this analysis.

The impact of these predictions on student performance in the end-of-semester written examinations was an average absolute increase of +2.67% weighted by participation across the modules with 3 of 10 modules recording a significant increase and 8 of 10 modules recording increased performance. This demonstrates

that we can use automated techniques to keep students informed of their progress on modules where there are a large number of students, and that when this is done then it will help students to progress.

Acknowledgements This research was supported by Science Foundation Ireland under grant number SFI/12/RC/2289, and by Dublin City University. The authors wish to thank Aisling McKenna for her help with the statistical analysis of some of the results.

References

1. K. E. Arnold and M. D. Pistilli. Course Signals at Purdue: Using Learning Analytics to Increase Student Success. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK '12*, page 267, New York, New York, USA, Apr. 2012. ACM Press.
2. M. D. Calvo-Flores, E. G. Galindo, M. C. P. Jiménez, and O. Pérez. Predicting students' marks from Moodle logs using neural network models. In *Proceedings of the IV international conference on multimedia and information and communication technologies in education (M-ICTEE2006)*, volume 1, pages 586–590, 2006.
3. K. Casey and J. P. Gibson. (m)Oodles of Data: Mining Moodle to understand Student Behaviour. In F. O'Riordan, F. Toolan, R. Hernandez, R. Smyth, B. Becker, K. Casey, D. Lillis, G. McGing, M. Mulhall, and K. O'Sullivan, editors, *Proceedings of the 10th International Conference on Engaging Pedagogy (ICEP10)*, pages 61–71, Maynooth, Ireland, Dec. 2010. Griffith College Dublin.
4. M. Cocea and S. Weibelzahl. Can log files analysis estimate learners level of motivation? In *LWA 2006. Lernen Wissen und Adaptivitt*, pages LWA 2006. Lernen–Wissensentdeckung – Adaptivität, Mar. 2006.
5. J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.
6. 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 E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
7. J. Platt et al. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*, 3, 1999.
8. C. Romero, S. Ventura, and E. García. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384, Aug. 2008.
9. A. Wolff, Z. Zdrahal, A. Nikolov, and M. Pantucek. Improving retention: Predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge - LAK '13*, page 145, New York, New York, USA, Apr. 2013. ACM Press.