Predicting Students' Final Exam Scores from their Course Activities

Michael Mogessie Ashenafi, Giuseppe Riccardi, Marco Ronchetti
Department of Information Science and Engineering
University of Trento
Trento, Italy
{michael.mogessie, giuseppe.riccardi, marco.ronchetti}@unitn.it

***Abstract*— A common approach to the problem of predicting students' exam scores has been to base this prediction on the previous educational history of students. In this paper, we present a model that bases this prediction on students' performance on several tasks assigned throughout the duration of the course. In order to build our prediction model, we use data from a semi-automated peer-assessment system implemented in two undergraduate-level computer science courses, where students ask questions about topics discussed in class, answer questions from their peers, and rate answers provided by their peers. We then construct features that are used to build several multiple linear regression models. We use the Root Mean Squared Error (RMSE) of the prediction models to evaluate their performance. Our final model, which has recorded an RMSE of 2.9326 for one course and 3.4383 for another on predicting grades on a scale of 18 to 30, is built using 14 features that capture various activities of students. Our work has possible implications in the MOOC arena and in similar online course administration systems.**

Keywords—automatic assessment; score prediction; peer-assessment;

I. INTRODUCTION

Automated prediction is a technique that has become prevalent in several fields and sectors including education, medicine, biology, politics, and finance. This prevalence is strongly attributed to recent advances in machine learning techniques.

Although the approaches adopted by prediction systems may vary, they all follow the same notion – make an *educated* guess about the value of a parameter by observing what variables affect that parameter and how they have affected it in the past. Ideally, the explanation that this guess is not random but educated is provided by the factoring of historical data about the variables and how they relate to the parameter into the prediction process.

The amount of data needed to make a *good* prediction depends on how complex the parameter being predicted is. That is, it depends on how many variables affect the value of the parameter. In reality, parameters to be predicted are fairly complex and large amounts of data are usually required to build decent prediction systems.

The availability of data does not necessarily guarantee that the prediction will perform well. Modelling the parameter to be predicted by identifying the variables and weighing their impact is a challenging task essential to the realisation of a successful prediction system.

In higher education, such systems have been used to predict the intermediate and final scores of students at different levels. Timely prediction of scores facilitates early intervention to identify students that may require special supervision and can be used to provide students with progress feedback.

Automated score prediction could also have a significant implication in the Massive Open Online Courses (MOOC) arena. Predicting the performance of students could help provide early insight into the attrition rates of courses administered in MOOC format. Such early indication would allow MOOC providers to explore corrective measures accordingly in order to increase the retention rates of their courses, as the majority of today's MOOCs suffer from immense attrition rates [10][18].

Student assessment techniques such as standardised tests and exams are able to obtain information about specific traits of students at a certain point in time. Gathering information about students that could explain their progress requires continuous recording of their activities using more sophisticated techniques. If designed well, such techniques could capture data that explain how students behave, communicate, and participate in learning activities and have the potential to predict how they would perform on end-of-course exams.

One practice that engages students in activities intended to improve their learning by evaluating other students' work is peer-assessment. Topping [24] defines peer-assessment as "…an arrangement in which individuals consider the amount, level, value, worth, quality, or success of the products or outcomes of learning of peers of similar status."

The reliability and validity of peer-assessment as either a formative or summative assessment tool has been studied in detail. Although there is agreement over the perceived values it brings to students and teachers, uncertainties remain regarding its use as an effective practice. A meta-analytic review by Falchikov and Goldfinch [25] and a critical analysis of peer-assessment studies by Topping [26] reveal these uncertainties.

Regardless of these uncertainties, peer-assessment provides a rich platform to gather significant information about students as they engage in assessing the works of their peers. In this study, we use an online peer-assessment framework to gather such information about students, which we then use to build our prediction model. Students participate in online peer-assessment task by submitting questions on pre-specified topics, by answering questions from their peers, and by rating their peers' answers.

In this paper, we present a linear regression model that utilises data generated by the activities of students in two courses to predict their final exam scores. This paper is organised as follows: In section II, we discuss several peer-assessment systems that are currently used in education in order to provide a comparative view of our peer-assessment framework and review previous work in score prediction. We then briefly discuss our peer-assessment framework and describe our prediction model in section III. In section IV, we provide details of the experiments and the results we obtained. We conclude our discussion in section V with a review of our work and our plans for the future.

II. Previous Work

A. Peer-Assessment

Although they may differ in the techniques they use or their overall design, all peer-assessment methods involve the practice of having students evaluate the works of their peers. Peer-assessment methods have been in use in education and other institutions for decades. See [13] for a detailed review of peer-assessment tools. Here, we discuss four peer-assessment platforms that we believe are relevant to our work.

PRAISE (Peer Review Assignments Increase Student Experience) is a generic peer assessment tool that has been used in the fields of computer science, accounting and nursing [3]. It has been used in introductory programming courses by students coming from different disciplines. Before distributing assignments, the instructor will specify criteria. Once assignments are distributed, students review details of each assignment and submit their solutions. The system waits for the number of submissions to reach a specified number and assigns review tasks to students. Students then review the solutions of their peers according to the criteria. Once all reviews for a solution are complete, the system checks if all reviewers agree according to the criteria and suggests a mark based on the criteria. If there is a disagreement among reviewers, the system submits the solution to the instructor for moderation. The instructor then needs to decide a mark and confirm the release of the result before a student can see their overall mark for the assignment.

PeerWise is a peer assessment tool, which students use to create multiple-choice questions and answer those created by their peers [4]. When answering a question, students are also required to rate the quality of the question. They also have the option comment on the question. The author of a question may reply to a comment that has been submitted by the student who rated the question.

PeerScholar is another peer-assessment tool that was initially designed for an undergraduate psychology class. It aims to improve writing and critical thinking skills of students [17]. First, students submit essays. Next, they are required to anonymously assess the works of their peers, after which they have to assign scores between 1 and 10, and write a comment for each of their assessments. Students are also allowed to rate the reviews they have received.

Workshop is a peer-assessment module for the Moodle E-Learning platform that lets students view, grade and assess their work or that of their peers [16]. The instructor co-ordinates and controls the assessment phases and is able to monitor the involvement of each student in each task. The instructor also has the ability to specify the criteria for computing grades and is also able to give different weights to different questions. The tool also allows assigning separate grades to submission of answers and assessment of submitted answers.

B. Score Prediction

The prediction of certain traits of individuals and groups from data generated by social networks and other platforms has been explored in several sectors. However, the vast majority of studies that relate to prediction of performance of students have had a particular focus on either computer science or computer literacy courses.

One early study conducted by Alspaugh [1] uses test results from three standardised tests – Thurstone Temperament Schedule [21], IBM Programmer Aptitude Test [14], and the Watson-Glaser Critical Thinking Appraisal [22] – and concludes that students who possess second level college calculus skills, have low levels of impulsiveness and sociability, and high reflectiveness have a good aptitude for computer programming.

Several studies have also investigated factors that can be used to predict the final scores of students. Fowler & Glorfeld [7] build a logistic classifier based on students' current GPA, math skills, SAT scores, and students' ages, which classifies students as having high or low aptitude for programming. The model is built using data from 151 students, 122 (81%) of which it classifies correctly.

Evans & Simkin [5] use six outcome variables as measures of computer proficiency – homework scores, scores in a BASIC programming exam with 19 matching questions, scores in a BASIC programming exam with 15 fill-in-the-blank questions, and first and second midterm scores. 49 predictor variables grouped into four categories – demographic, academic, prior computer training and experience, and behavioural – were used in building the models. A stepwise multiple regression model was built for each of the six predicted variables. The performance of these models was reported in terms of the coefficient of determination (R^2), with the model predicting homework scores having the highest value of 0.23.

Wilson and Shrock [23] conducted a study involving 105 students to predict success in an introductory college computer science course by examining twelve factors. Among these, three factors – comfort level, math skills, and attribution to luck for success – were found to be more important in

predicting mid-term scores. The performance of the linear model was reported to have an R^2 value of 0.4443.

As discussed above, most previous work in predicting the performance of students focused on very similar factors for making such predictions. Of these factors, the most common were math skills, high school scores, and standardised test scores.

Recent work has sought to exploit other more latent factors to predict success in computer science courses. Keen & Etzkorn [11] have built a model for predicting the average test scores of students of a computer science course by observing the buzzword density (the ratio of computer science related words to the total number of words) in the teacher's lecture notes. The intuition that higher buzzword density would imply more complex lecture notes and would, as a result, lead to lower average scores was supported by a strong negative correlation of -0.521 between buzzword density and average scores.

A recent study by Fire et al. [6] investigates the impact of interactions among students on their success in computer science courses as well as the correlation between students' scores. The study uses data from 163 students and applies graph theory and social network analysis techniques to predict students' final test scores and final grades. The features used for predicting students' final test scores include personal information features such as assignment scores and students' departments as well as topological features such as students' number of friends in the social network, which is built from homework assignment data and website logs, and friends' scores. Using these data, a single linear regression model is built to explore relationships among students. Another multiple linear regression model with stepwise inclusion is built to predict whether a student would score below 60, the passing mark for the course. The multiple regression model produces an R^2 value of 0.174 and Mean Absolute Error of 10.377.

Performance prediction has also been applied in MOOCs. One study uses students' performance on assignments from the first week together with their activity in the discussion forums and their peer-assessment task completion rate to build two logistic regression models that predict whether students will earn certificates of completion and whether they will achieve distinction, with accuracy levels of 79.6% and 92.6%, respectively [9].

Another study uses student behaviour data in a course administered in a MOOC format to predict whether a student will provide the correct answer for an in-video question at the first attempt [2]. Summary quantities such as the fraction of the video played and the number of pauses are extracted from clickstream data for each video-student pair and used to predict the likelihood of a student correctly answering questions in that video at the first attempt.

Automated prediction in MOOCs has however focused on early prediction of attrition rates from student behaviour. See [12], [19], and [20] for such studies.

III. Building The Prediction Model

A. The Peer-Assessment System,Participation, and the Data

The prediction model was built on data that were generated from the activities of students enrolled in two undergraduate level computer programming courses, Informatica Generale I (IG1) and Programmazione II (PR2), at the University of Trento in Italy. The central mechanism of the data collection required students to participate in a set of peer-based online homework activities throughout the course.

The online homework activities were carried out using a web-based peer-assessment platform that we built specifically for this purpose. The homework activities included three main tasks – *Ask A Question*, *Answer A Question*, and *Rate Answers*. Every week during the course, students would ask questions about topics that had been discussed in class, answer other students' questions, and vote for answers submitted by other students. They would also rate the levels of interestingness, relevance, and difficulty of questions.

The week starts with the teacher assigning the 'Ask A Question' task to all students, in which students submit questions regarding topics specified by the teacher that had already been discussed in class the previous week. After the deadline for completing the task has passed, the teacher filters the questions and selects a subset that will be used in the next task. The peer-assessment process is designed to obtain at least four answers to each question. Hence, the system recommends the number of questions to be selected by the teacher, taking into account the number of students participating.

The teacher then assigns the 'Answer A Question' task to all students. The system handles random assignments of the selected questions. It also guarantees that students will not be asked to answer their own questions and that each student is assigned only one question. When submitting their answers, students rate the difficulty, relevance, and interestingness of the questions on a scale of 1 (lowest) to 5 (highest).

The last task of the cycle is the 'Rate Answers' task, which asks students to rate the correctness of answers provided by students for the questions that had been selected on a scale of 0 (incorrect) to 5 (excellent). During assignment of the tasks, the system guarantees that the student that had asked the question is asked to rate the answers for their question. The system also guarantees that each question-answer set is assigned to at least four students and that the assignment of the tasks remains random for those who had not asked the questions that were selected by the teacher.

The web-based peer-assessment system was accessible through the Internet and students could complete their tasks at a location of their choice. Anonymity was preserved as students did not know whose question they had answered and whose answer they had rated or vice versa. All student activity including time of completion of tasks was logged by the system. Details of the design and implementation of an earlier version of the peer-assessment system are discussed in [15].

Because participation in the online peer-assessment activities was optional, some students did not participate at all while others opted out at several points during the course.

The university's exam policy permits students to withdraw from an exam without having their work assessed. Usually, students who expect to score lower than the minimum passing mark, 18 out of 30, either withdraw from or do not sit the exam, which, depending on the course, was either oral or written.

Although it is still possible to fail an exam, all students whose data were used to build the prediction model had passed their exams. The implication of this is that the model could not predict grades below 18 and was not able to predict dropping out. It is possible that future editions of the courses will record failing students, whose data can then be used to train the model to make such predictions.

Students also have the option to sit an exam in any of the five sessions available in an academic year. Therefore, although some students participated in the online peer-assessment activities, their final grades were not available as they had not sat their exams at the time of this experiment.

Consequently, although a total of over 400 students participated in the online homework activities for the two courses together, data from only 206 students were used in our experiment.

B. Preliminary Investigation

One of the online peer-assessment tasks requires students to evaluate a set of answers provided by other students for a question by assigning votes to the answers. At the end of the course, the number of votes a student has earned for all their answers will, among other measures of activity, constitute the overall degree of performance of the student in the online homework activities.

An intuitive approach to predicting the final scores of students using such data would be to explore the relationship between the number of votes a student has earned for their answers throughout the course and their final exam score.

This final exam score is represented as a whole number ranging from 18 to 30. We were not certain about finding a strong relationship, however, as these votes are assigned by students themselves and may, as a result, be inconsistent and inaccurate due to several factors such as inexperience of students in evaluating answers. A preliminary investigation of the existence of such a relationship and its strength would then be necessary to address this uncertainty.

We carried out this investigation by clustering students according to the number of votes they earned, which ranged from 0 to 21, and by computing the average final score for each cluster. We observed a rather weak relationship. We found that a linear fit hardly captured any relationship and that, although a better fit, a fourth degree polynomial was not an ideal model either. Attempting to model this relationship with polynomials of higher degrees would have eventually led to over-fitting.

This led us to conclude that student votes alone would not be strong predictors of final exam scores. We therefore decided to proceed with exploring more parameters that would explain the performance of students such as the amount of tasks they completed and the perceived level of difficulty of the questions they provided answers for.

C. Features of the Prediction Model

Our initial investigation explored 7 parameters in order to build a linear regression model. 16 additional parameters, most of which were computed from the initial 7 parameters, were later used to create more models. In favour of brevity, only a list of the parameters of the final model is presented below.

Tasks Assigned (TA) – The number of tasks that were assigned to the student

Tasks Completed (TC) – The number of tasks that the student completed

Questions Asked (QAS) – The number of 'Ask a Question' tasks the student completed

Questions Answered (QAN) – The number of 'Answer a Question' tasks the student completed

Votes Cast (VC) – The number of 'Rate Answers' tasks the student completed

Questions picked for answering (QP) – The number of the student's questions that were selected by the teacher to be used in 'Answer A Question' tasks

Votes Earned (VE) – The number of votes the student earned for their answers

Votes Earned Total Difficulty (VED) – The sum of the products of the votes earned for an answer and the difficulty level of the question, as rated by students themselves, for all answers submitted by the student

Votes Earned Total Relevance (VER) – The sum of the products of the votes earned for an answer and the relevance level of the question, as rated by students themselves, for all answers submitted by the student

Votes Earned Total Interestingness (VEI) – The sum of the products of the votes earned for an answer and the interestingness level of the question, as rated by students themselves, for all answers submitted by the student

Selected Q total difficulty (SQD) – The sum of the difficulty levels of the student's questions, as rated by students themselves, which were selected to be used in subsequent tasks

Selected Q total relevance (SQR) – The sum of the relevance levels of the student's questions, as rated by students themselves, which were selected to be used in subsequent tasks

Selected Q total interestingness (SQI) – The sum of the interestingness levels of the student's questions, as rated by students themselves, which were selected to be used in subsequent tasks

Number of Attempts (NA) – The number of attempts the student made to pass the course

D. The Prediction Model

The data were normalised using min-max normalisation, which converts the values of each parameter into a value between 0 and 1. We used the Weka data-mining toolkit [8] to build three sets of linear regression models – one for each course and an additional set using the combined dataset from both courses. First, we built models using the initial 7 features. We then built more complex models by adding a set of the computed features step by step. As a result, 3 sets of 7 linear regression models each were built.

Each model was tested using 10-fold cross-validation. The Root Mean Squared Error (RMSE) of the models was used for performance evaluation.

The model with the least RMSE was built using the 14 features discussed in III.*C*. The final score prediction model **m** is given by:

$$\boldsymbol{m(i) = \mathbf{C}^{\mathrm{T}}\mathbf{S}_i + 27.8967} \qquad \textbf{(1)}$$

, where **S** is a *14-by-n* matrix built from the 14 parameter values for *n* students, $\mathbf{S}_i$ is the i^{th} column in **S** representing student *i*, $\mathbf{C}^{\mathrm{T}}$ is the transpose of the column vector **C** given by,

$$\mathbf{C} = \begin{bmatrix} -3.9796 \\ -0.3193 \\ 0.6285 \\ 0.6744 \\ -2.1579 \\ 0.0968 \\ 0.7127 \\ 22.92 \\ -16.2889 \\ -5.0239 \\ 4.5355 \\ -3.7116 \\ 0 \\ -4.4189 \end{bmatrix}, \text{ and } \mathbf{S}_i = \begin{bmatrix} \mathrm{TA}i \\ \mathrm{TC}i \\ \mathrm{QAS}i \\ \mathrm{QAN}i \\ \mathrm{VC}i \\ \mathrm{QP}i \\ \mathrm{VE}i \\ \mathrm{VED}i \\ \mathrm{VER}i \\ \mathrm{VEI}i \\ \mathrm{SQD}i \\ \mathrm{SQR}i \\ \mathrm{SQI}i \\ \mathrm{NA}i \end{bmatrix}, \text{ for } i = 1, \ldots, n.$$

As can be observed from the column vector C, the model rewards students for earning votes for answering difficult questions as well as for asking challenging questions.

We believe that the two features that capture this information, VED and SQD, are good discriminators among students of different performance levels. This characteristic of the model is also coherent with the manner in which a teacher would award points to students.

I. Experiments and Results

A. Testing with Unseen Data

The final model was built on data from the IG1 course only. This provided us with the opportunity to test how the model would perform on data coming from another course. We thus tested the model with 101 instances from the PR2 course. The RMSE was found to be **3.4383**. This result is encouraging, taking into account the fact that the test data come from a different course. Although the levels of the two courses were different and were attended by different groups of students, the prediction errors of the system were comparable when predicting performance of students attending the PR2 course using data from the IG1 course.

B. Is the Prediction any Better than Random Guessing?

In order to determine if our prediction model outperformed random assignment techniques, we developed several mechanisms of random guessing. First, we assigned a grade to each of the 206 students by randomly selecting a number from the valid range of grades, 18 to 30. We performed this random assignment 10000 times and evaluated the average RMSE of these assignments. We performed this grade assignment for students of both courses IG1 and PR2. The average RMSE for this technique was computed as **5.0370.**

We then performed a systematic random assignment of grades by sampling from a prior distribution of grades from previous editions of the courses. As shown in Fig. 1, the distributions of grades for the two courses are not similar. We therefore decided to sample grades from separate prior distributions, one for each course.

After exploring several probability distributions, we found that the best possible fits for both courses were kernel distributions. Fig. 2 shows these distributions plotted against the probability density function plots of the two courses.

For each course we sampled from its respective distribution and assigned grades to students. We performed this assignment 10000 times, averaged the RMSEs of all assignments, and obtained an average RMSE of **4.9405** for IG1 and **5.0122** for PR2.

Finally, we sampled from the prior distribution of grades by assigning an index to each grade and by randomly selecting a number from these indices, which formed a uniform distribution. This would allow us to include instances which would not be captured by the kernel fits shown in Fig. 2.

We performed this sampling for the two courses separately. As before, we performed the random assignment 10000 times and averaged the RMSEs, obtaining an average RMSE of **4.8869** for IG1 and **4.9632** for PR2. Tables I and II provide a comparison of the random assignment techniques and our model. As our model is tested with the respective data for each course, there is a variation, albeit slight, in its performance on data from the two courses.

The histograms in Fig. 3 show the prediction errors. In an ideal prediction model, errors would be significantly low. Hence, the histogram of such a model would have a slender shape, with its peak near the centre of the horizontal axis.

Although our model outperforms all random assignment techniques, its strength is evidenced in how it outperforms assignment techniques even when random assignment is aided

by information from previous editions of the courses to reduce the frequency of assigning grades that were not common. The strength of our model is also reflected by the fact that none of the 10000 assignments for each technique scored an RMSE lower than that of the model.

TABLE I. EVALUATION OF PREDICTION METHODS FOR IG1

Prediction Method	RMSE
Sampling from a uniform distribution	5.0370
Sampling from a kernel distribution	4.9405
Sampling from previous scores directly	4.8869
Linear regression model	**2.9326**

TABLE II. EVALUATION OF PREDICTION METHODS FOR PR2

Prediction Method	RMSE
Sampling from a uniform distribution	5.0370
Sampling from a kernel distribution	5.0122
Sampling from previous scores directly	4.9632
Linear regression model	**3.4383**

II. FROM PREDICTING SCORES TO PREDICTING GRADES

It would be interesting for teachers to groups students into several categories based on their performance. While scoring on a scale of 18 to 30 may be too broad a range to provide such information, grades, numerical or otherwise, provide this functionality. Such grading systems may also be fine-tuned to decide the granularity of these groups.

Here, we transform the scores of students on an 18 to 30 scale to numerical grades on a scale much similar to the A to F grading system. Numerical grades range from 0 to 4, with a grade of 4 corresponding to an A, 3 to a B, and so on. Scores are converted to numerical grades by assigning a single grade to a range of scores. Hence, scores 28 to 30 will be assigned a grade of 4, 25 to 27 a grade of 3, 22 to 24 a grade of 2, 18 to 21 a grade of 1 and those below 18 a grade of 0.

In order to perform this experiment, we used the same features explained in section III.C, to build a new model that predicts the numerical grade of a student using the newly transformed data. As before, we used 10-fold cross-validation to evaluate the performance of our model. The model that was built using data from the IG1 course performed better than that built on the PR2 course, albeit slightly. We therefore report the performance of the winning model only.

Although it might seem that this way of predicting grades is a 5-class classification problem, the fact that the input variables assume continuous values makes the classification task impossible. Indeed, classifiers such as Naïve Bayes Classifier, Decision Trees, and Logistic Regression have prohibitively poor performance.

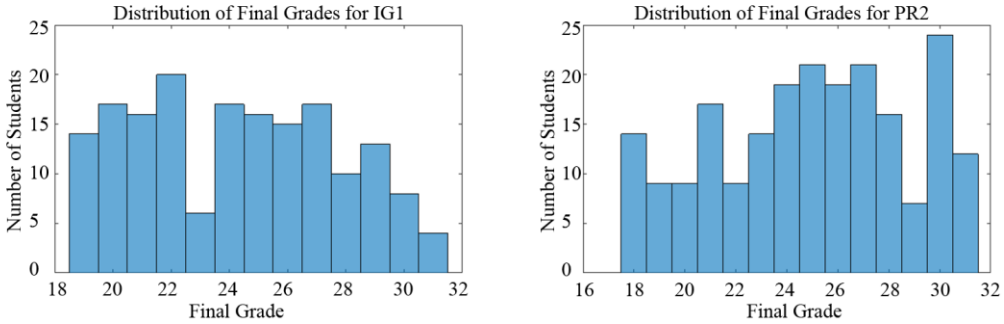

Figure 1. Distributions of grades for IG1 (left) and PR2 (right)

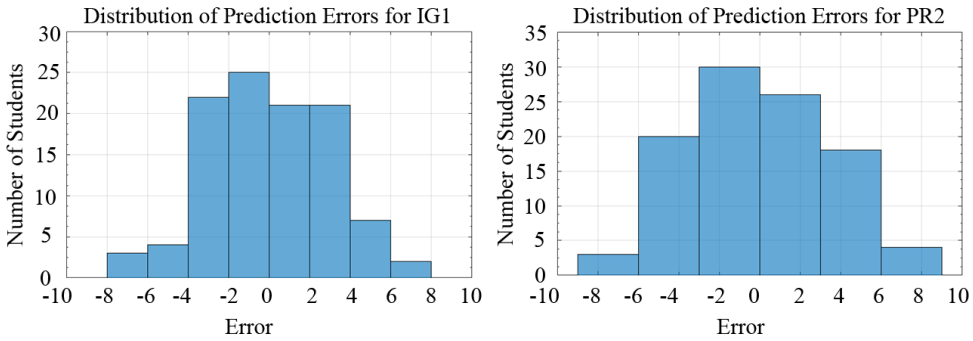

Figure 3. Histograms of the prediction errors for IG1 (left) and PR2 (right)

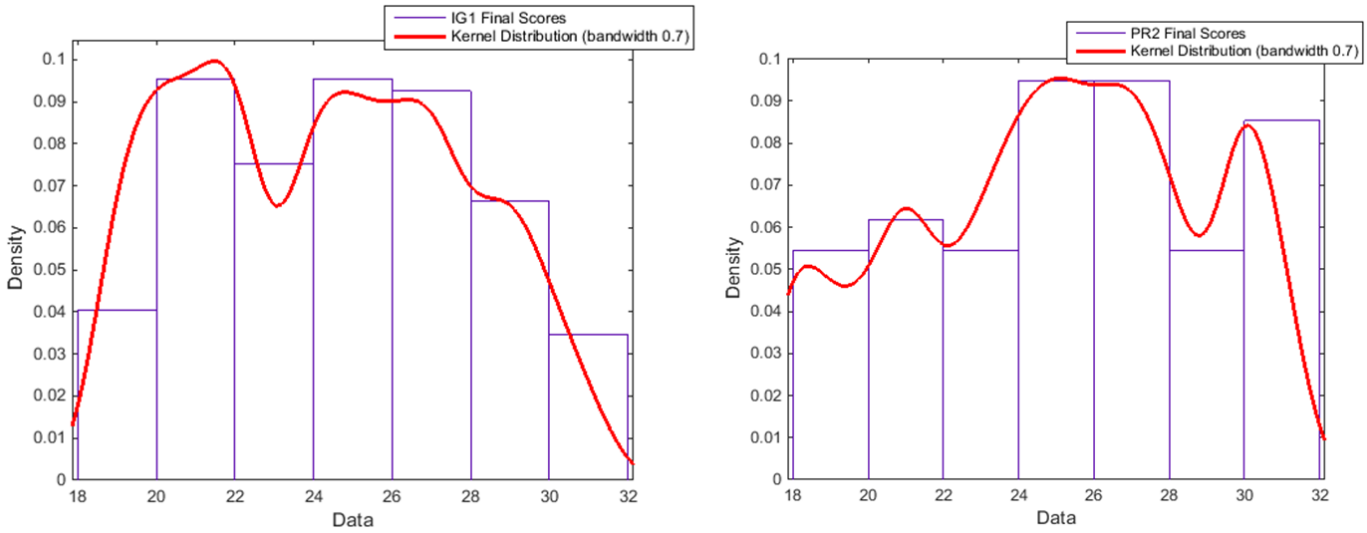

Figure 2. Histograms of the final scores of students of IG1 (left) and PR2 (right) plotted against kernel distributions

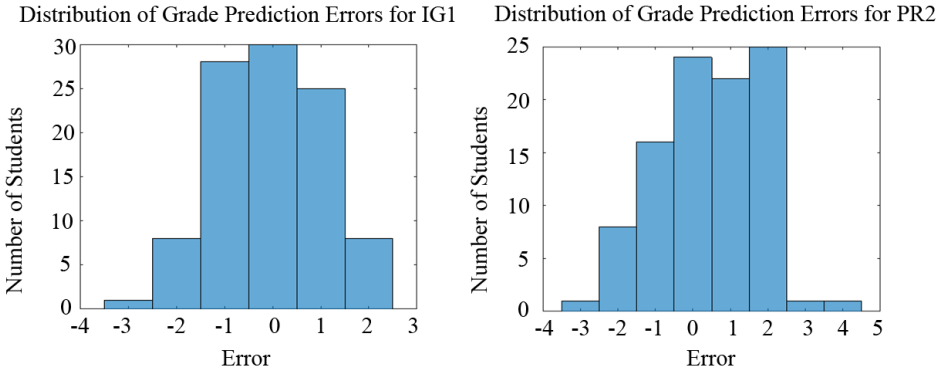

Figure 4. Histograms of the grade prediction errors for IG1 (left) and PR2 (right)

Although it might seem that this way of predicting grades is a 5-class classification problem, the fact that the input variables assume continuous values makes the classification task impossible. Indeed, classifiers such as Naïve Bayes Classifier, Decision Trees, and Logistic Regression have prohibitively poor performance.

Linear regression, on the other hand, can handle such data as the grades are still numerical. However, the prediction values are continuous and do not necessarily map into one of the five grades. We therefore use a function that rounds prediction values to the nearest integer to make the prediction valid. As a result, the RMSE of the rounded predictions, which is slightly higher than the RMSE computed on the actual predictions, is reported.

The winning model scored a 10-fold cross-validation RMSE of 1.1225, a significant decrease from the previous value of 2.9326 when predicting numerical scores. When tested on unseen data from the PR2 course, the model scored a much lower RMSE of 1.4428 than the previous score of 3.4383. The prediction errors of this model for both courses are depicted in the histograms in Fig. 4.

In the following tables, we report comparisons between our model and baselines constructed in the exact manner as before.

Here, we sample from normal distributions instead of kernel distributions as they fit better the grade distributions for both courses.

TABLE III. EVALUATION OF GRADE PREDICTION METHODS FOR IG1

Prediction Method	**RMSE**
Sampling from a uniform distribution	1.8299
Sampling from a normal distribution	1.6760
Sampling from previous grades directly	1.5336
Linear regression model	**1.1225**

TABLE IV. EVALUATION OF GRADE PREDICTION METHODS FOR PR2

Prediction Method	RMSE
Sampling from a uniform distribution	1.8604
Sampling from a normal distribution	1.6485
Sampling from previous grades directly	1.5267
Linear regression model	**1.4428**

Table V shows the performance of the grade predictor in terms of accuracies. For IG1, 83% of its predictions fall within the range 0 to 1 grade point difference whereas for PR2, it performs less, with 63% of its predictions falling in the same range.

TABLE V. PREDICTION ACCURACIES OF THE MODEL

Course	Exact Prediction	Within 1 Grade Point	Within 2 Grade Points
IG1	0.30	0.83	0.99
PR2	0.24	0.63	0.97

The grade prediction model m is given by:

$$m(i) = \mathbf{C}^{\mathbf{T}}\mathbf{S_i} + 5.75 \qquad (2)$$

, where S is a 14-by-n matrix built from the 14 parameter values for n students, S_i is the i^{th} column in S representing student i, C^T is the transpose of the column vector C given by,

$$\mathbf{C} = \begin{bmatrix} -0.1593 \\ 0.0038 \\ 0.0642 \\ -0.0193 \\ -0.0149 \\ 0.0499 \\ -0.4667 \\ 6.2881 \\ -4.9538 \\ 0.1553 \\ 2.1008 \\ 1.5545 \\ -3.5342 \\ -0.2977 \end{bmatrix}, \text{ and } \mathbf{S}_i = \begin{bmatrix} \text{TA}i \\ \text{TC}i \\ \text{QAS}i \\ \text{QAN}i \\ \text{VC}i \\ \text{QP}i \\ \text{VE}i \\ \text{VED}i \\ \text{VER}i \\ \text{VEI}i \\ \text{SQD}i \\ \text{SQR}i \\ \text{SQI}i \\ \text{NA}i \end{bmatrix}, \text{ for } i = 1, \ldots, n.$$

Similar to the previous model, this model rewards students who earn votes for answering questions that are regarded as difficult and interesting, as well as for asking questions which are challenging and relevant.

III. DISCUSSION AND CONCLUSIONS

Performance prediction in educational activities has been studied before. Most previous studies, however, were limited to analysing previous performance information of students to make such predictions. Most of this information came from high school level performance data and college entrance examination scores.

Today, students themselves generate significant amounts of data throughout their studies. The major goal of our work was to take advantage of such information in order to predict student performance. In this paper, we presented a linear regression model for predicting final exam scores of students by observing data that are generated from their online course activities. We implemented our web-based peer-assessment system in two courses and used the data from the system to build our model. The preliminary results of our prediction model are encouraging.

We believe the techniques and settings we used to generate data about students and make predictions about their performance are novel. Although work in predicting student success has recently gained more focus, most of the attention has been directed towards predicting attrition rates in courses administered online, specifically MOOCs.

However, dropout is not a problem specific to MOOCs. For instance, a 2013 report authored for the European Commission [27] states that the university completion rate in Italy for students from lower socioeconomic backgrounds, ethnic minorities, people with disabilities, or adult learners is as low as 46% and attributes this to lack of attention to diverse student populations and lack of student-centred approaches to designing educational programmes. Studies in close supervision of students and early detection of at-risk individuals are therefore in order and we believe that predicting student performance in higher education settings and in mandatory courses at the undergraduate level is an important part of such studies.

US-style grading techniques help group students into several performance groups and predicting grades instead of scores on longer ranges aids in this aspect. We therefore built a similar regression model for predicting the grades of students. The prediction errors are much lesser for this model and it can predict, within a grade point range, the grades of the large majority of students, as seen in table V.

The general assumption behind predicting students' final exam scores from peer-assessment data is that students' performance in the online peer-assessment tasks would be consistent with their performance in the final exams. This assumption in our particular setting may, however, not always hold true across the class because the reward for participating in the online assessment tasks, which are not mandatory, is not commensurate with the reward for performing well in the final exam. Such exceptions would explain the positive errors of the prediction.

It is also possible that some students who participated well in the activities may have performed poorly in the final exam for unprecedented reasons. This would account for the negative errors of the prediction. Whether a significant increase in the amount of data used to build the model would improve its performance is yet to be seen. We plan to conduct new experiments using the data we will gather in the next two semesters.

In the experiments we conducted, prediction of scores was made after courses had ended and before students had sat final exams. However, prediction is only effective when it is done in a timely manner and is no good if it provides important information about students at a time when little can be changed

to help them improve. In an upcoming study, we intend to apply the prediction model discussed in this work to student performance data as it is generated in order to make predictions on a weekly basis to facilitate early detection and supervision of students that may require special attention.

As more and more higher education institutions make their courses available for learners through platforms such as Massive Open Online Courses (MOOCs), the immense amount of data generated make it possible to provide continuous and automated assessment of student progress.

The peer-assessment framework that we use is complementary to traditional classroom lessons. Nonetheless, the prediction system is not tied to the pedagogy. This makes it easy to extend this approach of student supervision and assessment to non-traditional learning environments such as flipped classrooms.

We are hopeful that, although the framework we utilised in this study is not very similar to MOOCs, the way we build a prediction system on top of the data can be adopted by MOOCs and similar platforms. There are already studies that use such data to predict attrition rates of courses administered in a MOOC format but we have demonstrated in this study that it is possible to go further and learn students' trends as they participate in courses to provide timely supervision from such rich data.

REFERENCES

[1] Alspaugh, C. A. (1972). Identification of some components of computer programming aptitude. Journal for Research in Mathematics Education, 89-98.

[2] Brinton, C. G., & Chiang, M. MOOC Performance Prediction via Clickstream Data and Social Learning Networks.

[3] De Raadt, M., Lai, D., & Watson, R. (2007, November). An evaluation of electronic individual peer assessment in an introductory programming course. In Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88 (pp. 53-64). Australian Computer Society, Inc..

[4] Denny, P., Hamer, J., Luxton-Reilly, A., & Purchase, H. (2008, September). PeerWise: students sharing their multiple choice questions. In Proceedings of the fourth international workshop on computing education research (pp. 51-58). ACM.

[5] Evans, G. E., & Simkin, M. G. (1989). What best predicts computer proficiency?. Communications of the ACM, 32(11), 1322-1327.

[6] Fire, M., Katz, G., Elovici, Y., Shapira, B., & Rokach, L. (2012). Predicting student exam's scores by analyzing social network data. In Active Media Technology (pp. 584-595). Springer Berlin Heidelberg.

[7] Fowler, G. C., & Glorfeld, L. W. (1981). Predicting aptitude in introductory computing: A classification model. AEDS Journal, 14(2), 96-109.

[8] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

[9] Jiang, S., Warschauer, M., Williams, A. E., O'Dowd, D., & Schenke, K. (2014). Predicting MOOC Performance with Week 1 Behavior. In Proceedings of the 7th International Conference on Educational Data Mining.

[10] Jordan, K. (2013). MOOC Completion Rates: The Data, Available at: http://www.katyjordan.com/MOOCproject.html [Accessed: 07/01/15].

[11] Keen, K. J., & Etzkorn, L. (2009). Predicting students' grades in computer science courses based on complexity measures of teacher's lecture notes.Journal of Computing Sciences in Colleges, 24(5), 44-48.

[12] Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014). Predicting MOOC Dropout over Weeks Using Machine Learning Methods. EMNLP 2014, 60.

[13] Luxton-Reilly, A. (2009). A systematic review of tools that support peer assessment. Computer Science Education, 19(4), 209-232.

[14] McNamara, W. J., & Hughes, J. L. (1969). Manual for the revised programmer aptitude test. White Plains, New York: IBM.

[15] Mogessie, M., Riccardi, G., & Ronchetti, M. (2014, June). A Web-Based Peer Interaction Framework for Improved Assessment and Supervision of Students. In World Conference on Educational Multimedia, Hypermedia and Telecommunications (Vol. 2014, No. 1, pp. 1371-1380).

[16] Moodle Workshop Module - https://docs.moodle.org/28/en/Workshop_module

[17] Paré, D. E., & Joordens, S. (2008). Peering into large lectures: examining peer and expert mark agreement using peerScholar, an online peer assessment tool.Journal of Computer Assisted Learning, 24(6), 526-540.

[18] Parr, C. (2013). MOOC Completion Rates 'Below 7 %', Available at: http://www.timeshighereducation.co.uk/news/mooc-completion-rates-below-7/2003710.article. [Accessed: 07/01/15].

[19] Rosé, C. P., & Siemens, G. (2014). Shared task on prediction of dropout over time in massively open online courses. EMNLP 2014, 39.

[20] Sharkey, M., & Sanders, R. (2014). A Process for Predicting MOOC Attrition.EMNLP 2014, 50.

[21] Thurstone, L. L. (1949). Thurstone temperament schedule. Science Research Associates.

[22] Watson, G. (1980). Watson-Glaser critical thinking appraisal. San Antonio, TX: Psychological Corporation.

[23] Wilson, B. C., & Shrock, S. (2001, February). Contributing to success in an introductory computer science course: a study of twelve factors. In ACM SIGCSE Bulletin (Vol. 33, No. 1, pp. 184-188). ACM.

[24] Topping, K. (1998). Peer Assessment Between Students in Colleges and Universities. Review of Educational Research. http://doi.org/10.3102/00346543068003249

[25] Falchikov, N., & Goldfinch, J. (2000). Student Peer Assessment in Higher Education: A Meta-Analysis Comparing Peer and Teacher Marks. Review of Educational Research. http://doi.org/10.3102/00346543070003287

[26] Topping, K. J. (2010). Methodological quandaries in studying process and outcomes in peer assessment. Learning and Instruction, 20(4), 339–343. http://doi.org/10.1016/j.learninstruc.2009.08.003

[27] J. Quinn,"Drop-out and completion in higher education in Europe among students from under-represented groups," European Union, Oct. 17, 2013.