Capturing Student Interaction within Learning Management Systems (LMS) for Identification of At-Risk Students

ABSTRACT

Distance education offerings in the form of massive open online courses (MOOCs) and traditional universities have seen a surge in enrollments from students across the world. While enrollments are large it is not clear if these online offerings are able to achieve the desired learning outcomes as in the case of in-class face-to-face learning. Learning management systems (LMS) aid both online and in-class course offerings by providing content and collaborative tools between students and instructors. Learning analytics seeks to analyze the data extracted from LMS server logs to identify student learning behaviors and engagement characteristics to further help the students in achieving academic success. Analysis of these datasets can also help the instructor in designing improved course content, identifying common challenges across students and improve overall pedagogy. The objective of this study is to develop and assess machine learning methods that use features extracted from LMS server logs to perform early and real-time prediction of student performance within a course. Using this information the proposed approaches seek to identify students' at-risk of failing or dropping a class and provide timely feedback from instructor/advisor to keep the student on track. Leveraging data across multiple courses taken by a given student, the engineered features capture student interactions and course characteristics. We performed a comprehensive evaluation using the de-identified data obtained from Canvas Network open courses. Our experimental results show that we can predict the student final learning outcomes with high accuracy. On the basis of the gini index, we also help identify features found to be important towards final course performance during different stages of the course.

KEYWORDS

Early Warning, Learning Analytics, Regression, Classification, Early Feature, Student Behavior

ACM Reference format:

. 2019. Capturing Student Interaction within Learning Management Systems (LMS) for Identification of At-Risk Students. In *Proceedings of LAK'19: International Conference on Learning Analytics and Knowledge, Temple, Arizona, USA, March 7-9 (LAK'19),* 9 pages.

DOI: 10.475/123_4

LAK'19, Temple, Arizona, USA

DOI: 10.475/123_4

1 INTRODUCTION

With the advancement in learning technologies, education institutions increasingly rely on the online sources for delivering educational content and achieving learning outcomes [16]. Online or distance education can be synchronous i.e., in conjunction with a brick-and-mortar class happening at the same time, asynchronous or hybrid where the online material supplements traditional in-class material. Massive Open Online Courses (MOOCs) since their inception have promised the opportunity of delivering low-cost (free) educational resources to thousands of students across the world [20].

Both, brick-and-mortar educational institutions and MOOCs use learning management systems (LMS) or course management systems. Prime examples include Blackboard (blackboard.com), Canvas (canvas.net) and Moodle (https://moodel.org) for online access to course content. These systems allow for collaboration and communication amongst the different stakeholders within a course: (i) instructors, (ii) students and (iii) teaching assistants. The server logs serve as a source of student-interaction data with the LMS that can be used to identify student engagement and learning behaviors for a given course. Learning analytics researchers have developed several different approaches to analyze this interaction data for several purposes: (i) improving content and learning outcomes by identifying for the instructor, course content that several students face difficulty in mastering, (ii) predicting students' future academic performance to facilitate better degree planning and advising (iii) early identification of students who may be at risk of failing a class and would benefit from attention/intervention by course staff and (iv) identifying successful pedagogical approaches that helps students learn better.

Learning management systems utilized for MOOCs provide students and instructors with a collaborative way of overcoming the limitations of traditional classroom space while also saving time. Educators are now able to assess overall learning performance and determine the best ways for students to learn from the course while at the same time addressing any academic challenges they may be facing [2, 8, 14]. Using server logs, various student engagement and interaction features can be derived. Examples include the amount of time required for studying individual chapters, completing quizzes and wrapping-up assignments [20]. By evaluating these student interactions as well as the course information, learning analytics can identify patterns associated with student learning. Instructors, as well as other stakeholders provided with access to these data analytics approaches can identify if students are achieving the class learning goals in a timely manner and provide interventions and personalized feedback [11].

In this paper, we implement machine learning methods to identify students who are at risk of falling behind in a timely fashion. We simulate two real-world scenarios of students enrolled

within MOOCs. Specifically, we name these approaches as **Student-Specific** and **Course-Specific**. We seek to perform the task of in-class prediction i.e., using interaction data extracted from LMS server logs to predict the final grade for a student and identify students who are at risk of failing a course. Another key objective of our proposed methods is to identify these students earlier in the semester (also used interchangeably with the term). We also seek to identify the significant features that are strong predictors of identifying at-risk students. We evaluated the proposed methods on a Canvas [17] that is comprised of de-identified data from 376 Canvas Network open courses which are also MOOC offerings. Our results highlighted the strengths of the proposed approaches in predicting students who are at at-risk of not passing the class using features derived from LMS data.

2 LITERATURE REVIEW

According to the National Center for Education Statistics [15], around 41% of students who enrolled in a four-year undergraduate program in Fall 2009 failed to graduate within six years of that program. Schneider et al. [22] estimated that the hidden cost for college dropouts in a single academic year is $3.8 billion [25]. In order to improve the retention, several researchers have focused on the analysis and prediction of student's performance based on student's past learning related habits and aptitudes. Romero et al. [21] evaluated various data mining techniques to classify students as high and low performers based on their LMS usage data. Ren et al. [20] developed a multi-regression based model to predict the performance of a student per assessment (HW) based on student interaction data for several MOOCs. Devasia et al. [3] predicted students' performance best by analyzing student social features like that of gender and lifestyle habits. Instead of focusing on graded learning features like assignments and quizzes, Sahebi and Brusilovsky [23] took advantage of students' non-graded activity and found that their approach could reduce the error within student performance predictions.

Besides in-class performance prediction, an understanding of suitable approaches and theories of learning analytics is also required for further examination of learning behavior [12]. Pittman [18] compared data mining techniques used to predict student retention and found that logistic regression was the most suitable. Boroujeni and Dillenbourg [1] discovered some common study patterns based on the MOOC interaction sequence and found that these study pattern transitions probabilities correlated with different learners. Zhang and Rangwala [25] developed an Iterative Logistic Regression (ILR) method to address the challenge of early predictions and got a much more precise answer than results obtained from standard logistic regression.

In this paper, we study the application of machine learning as it relates to early in-class student grade prediction. Similar performance prediction techniques have been explored in different settings. Jiang et al. [9] used a combination of students' first-week assignment performances and social interactions within the MOOC to predict their final performance through logistic regression. He et al. [5] investigated the early warning signs of students at risk of failing a MOOC by evaluating multiple offerings under potentially non-stationary data. They built predictive models weekly based on the numerous offerings of a course. Jokhan et al. [10] designed an early warning system based on the students' features such as gender, age, social status and engagement features to achieve a 60.8% accuracy based on that particular model. Due to the absence of data from previous classes, Hlosta et al. [6] developed a 'self-learner' method which used current course data as the training set to identify the at-risk students.

Several prior studies [20] use MOOC server logs to predict homework grades or dropouts. In this paper, we seek to identify students' at-risk of failing a course by using LMS-derived features within standard machine learning models. Our key contributions stem from benchmarking and leveraging data across multiple courses (rather than a single course) for a given student and focusing on the early identification of at-risk students. We simulate two real world scenarios for in-class final performance prediction, first centered around students enrolled in multiple MOOCs and second involved developing course-specific models that assume multiple offerings for a given course across different terms.

3 PROBLEM DEFINITION

Given a database about the interaction of students with a learning management system for a given course, the objective of this study is to develop classification methods to identify students (early on) who will perform well in a target/current course. The set of interaction features capturing student engagement and learning habits extracted from the LMS is denoted by F_i^j for the i-th student and j-th course. Formally, the objective of the classifier is to learn a mapping function $f : F \rightarrow 0|1$ that takes as input the feature from the current class F_i^c and output 0 (representing passing a course) and 1 (representing failing a course). Additionally, the proposed algorithms seek to make these performance predictions early on to assist the student (who are at risk of failing) do better. As such, we assess the performance of the proposed algorithms by using features extracted from the first few days or weeks of the course. We encode this by extracting features only from the first 10%, 20%, 30% and 40% of the course during training and testing.

Figure 1 shows the student interaction data for a typical student. We can view students' various activities at different timestamps within the Figure. The Y-axis shows the number of requests made per day by the student. The dots along the time-series indicate specific course-related events i.e., submission of quizzes or assignments made by this student. The percentage value indicates the score earned by the student on the particular graded activity. Along the top, we highlight the feature extraction from the start of the semester based on the amount of time we want to consider. For the given course we show in Figure 1 X set to 0.1 indicates the first 10% features of the class *C1* will be used. In our study, we set X to a 10%, 20%, 30% and 40% to catch student features towards the beginning of the course, as we defined as **Early Stage Feature**.

We simulate two common real world scenarios centered around a student and course, described in detail below.

3.1 Student-Specific Approach:

In the *Student-Specific* approach, we simulate the real world scenario of students enrolling in multiple courses over time. Each enrollment record associated with a student-course pair is stored in a database

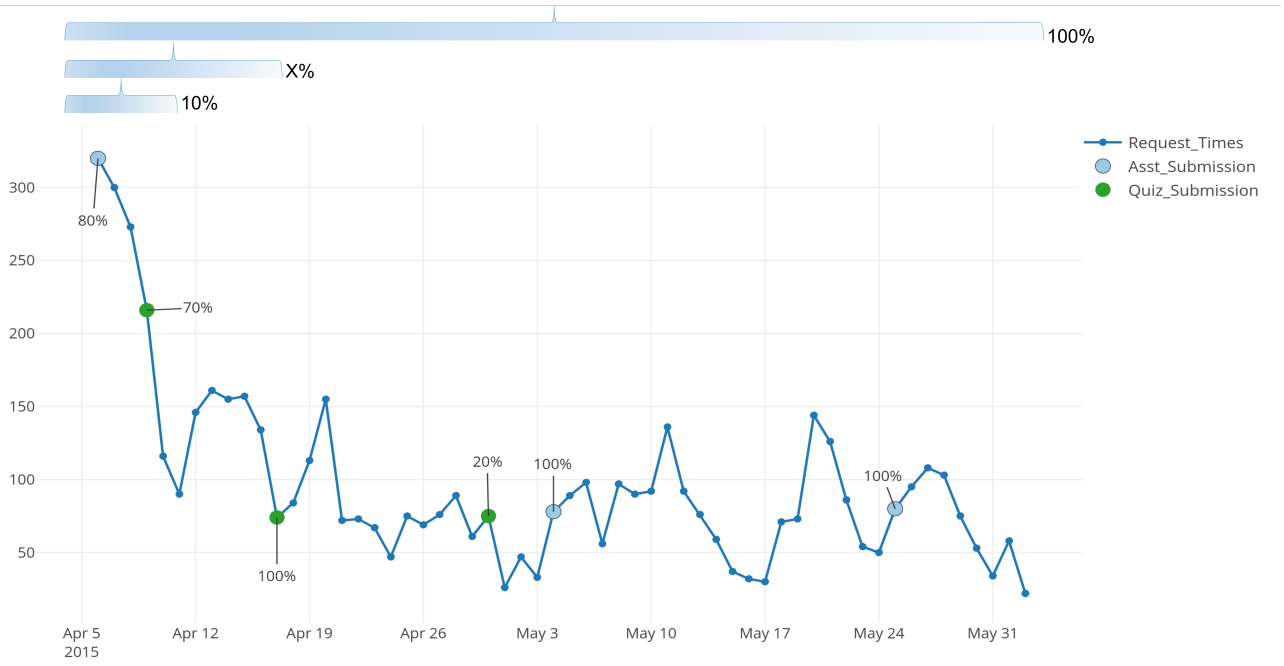


Figure 1: A sample student engagement time-series data

and we call it *Student-Courses records* . The graphical representation of student-Courses records is shown in Figure 2. We seek to predict the performance of a student within a given class based on performance/interaction within prior Student-Course records. Specifically, we are predicting the final grade of a student in a current or active class using interaction data from the first few weeks of the current/target class.

The graphical representation of this approach shown in Figure 3. We divide the Student-Course dataset into a training set and a testing set. The training set is regarded as the set of courses completed in the past and the test set is the set of active/current on-going courses. We split the data such that training set accounts for 90% of the dataset. We also seek to understand the relationship between prediction accuracy and amount of data in terms of time/weeks needed for deriving the features and hence the predictions. For the test set we predict within the first few weeks by setting the parameter X in Figure 3 to smaller values such as 10%, 20%, 30%, 40% to evaluate this early warning approach. We combine both student- and course-related features as described below for predicting the final grade. Lastly, we input both training and testing data into the three machine learning methods.

3.2 Course-Specific Approach:

Educational institutions usually offer the same course across different semesters. We also consider an alternate way of identifying possible at-risk students in a course by comparing student's performance in previously offered course (completed course). The graphical representation of the *Course-Specific* approach is shown

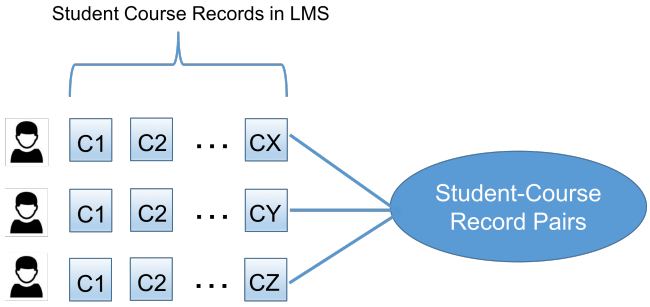


Figure 2: Student-Course Records

in Figure 4. Our dataset only has the course discipline information, and we cannot identify the previously offered courses in our dataset.

We simulate this by sampling the training and testing data from the same course. We use 10% of students as a testing set presented as $\{F_1^c,...,F_m^c\}$ in Figure 4. We assume this 10% of students would take this course next semester. The remaining students are training set and presented as $\{F_1^p,...,F_n^p\}$. For the Course-Specific Approach, we only use the student features because the course features (like CourseLen) would remain constant for the same class. Unlike the student-specific approach we train multiple course-specific models. We applied this approach to a total of 107 courses and averaged the accuracy along with the final scores for each experiment. The

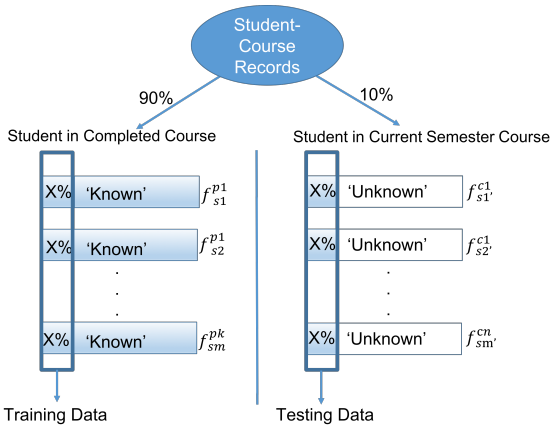


Figure 3: Student-Specific Approach

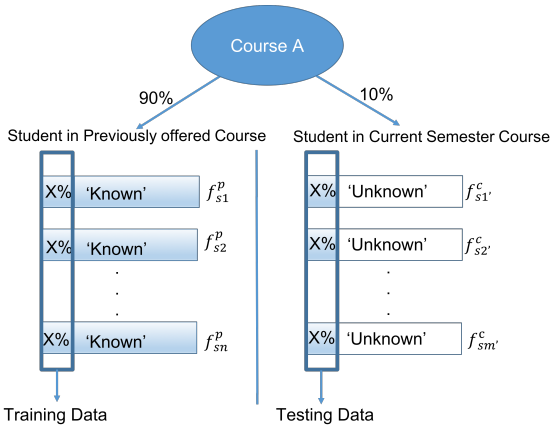


Figure 4: Course-Specific Approach

features extracted are similar for the student-specific and course-specific approach.

4 METHODS

We introduce several machine learning methods which are used in student performance analysis. We then provide detailed description on the student/course features we use in our model.

4.1 Machine Learning Method Description:

4.1.1 Logistic Regression: Logistic regression (LR) is a widely used machine learning method. In simplest terms we use logistic regression to predict a binary outcome such as good or bad [7]. The core of LR in its simplest form is a logistic or sigmoid function. It is described further in (1). The range of logistic functions existing between 0 and 1. Logistic regression results in probabilistic outputs and training can be done easily using a stochastic gradient based approach. The hypothesis function is described further in Equation (2). Formula 2 is the probability of y = 1 when we have a specific input x and a parameter θ . For our study the input x is the student interaction features.

$$g(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

$$P(y = 1|x; \theta) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2}$$

4.1.2 Random Forest: Random Forests (RF) are another widely used technique within machine learning. The core concept is to construct multiple decision trees at the time of training by sampling and tree pruning. The output of random forest is then an aggregate e.g., mode (most frequently occurring number) of individual trees. Such an approach is also considered as an ensemble classifier and is known to reduce the prediction errors when using multiple decision trees [4, 13]. It is important to note that even if we input the same features every time, we may get a different output based upon the non-deterministic randomness of RF.

4.1.3 k-nearest neighbors algorithm: The k-nearest neighbor's algorithm (k-NN) is a non-parametric method used for classification and regression [12]. Given a training dataset with N instances accompanied by a set of features unique to that training set. We identify for a test/prediction instance k closest neighbors using a similarity/distance function and predict the class label for the test instance by using the labels of the k neighbors. Prediction label is determined using a majority vote or weighted majority vote. In this study we used standard Euclidean distance and simple majority for the final decision.

4.2 Features Description:

To find the best possible learning related patterns for each student, we extract 13 features grouped into the following four categories: (i) course feature, (ii) quiz feature, (iii) assignment feature, (iv) access times feature. The description of each of these extracted features are as follows:

(i) Course Features: These features capture course statistics including discipline.

- **CourseLen**: denotes the total consecutive time duration of all course meetings. To define an early in-class feature, an experimental course must have a clear start and end date. This feature may be a good predictor if students adapt to a specific length of a particular course over another.

- **Type**: There were 12 different discipline courses in the dataset used in this study (detailed information listed in Table 1). We include this feature to capture a student's interest/aptitude within a specific discipline over another.

- **Size**: denotes the number of students enrolled for a given course. This feature will be a good predictor of if students tend to concentrate better within a smaller-sized class over one that is larger and more densely-packed.

(ii) Quiz Features: These features seek to capture performance of students based on quiz submissions and trials.

- **#Q**: denotes the total number of quizzes offered over one course. In our database, there are some quizzes and assignment with 0 possible_points. We did not include practice

quizzes having a raw score of 0 possible_points because practice quizzes bear no effect on course grades.

- **QSubmission**: is the number of quiz submissions made by a student before a given cut-off period. We normalize this feature value by comparing it to the average submission of the class. QSubmission is an important engagement feature in the course management system and could be a strong predictor. A passing student completes most quizzes and assignments on time.

- **QScore**: sums the raw scores of all quizzes taken by a student and is calculated from each quiz submission and then normalized by comparing to the average quiz score of that of the entire class. QScore is one of the most important feature aspect of grade prediction.

- **QAttempt**: is the average number of attempts of quiz submissions made by one student. Certain quizzes in our database allowed for multiple submissions and LMS retains the highest score. We believe this feature to be a success indicator since the more attempts made by a student indicate the willingness to learn and work hard.
- **QTime**: is the average time a quiz has remained opened before submission by one student. Based on the student, a longer quiz duration may indicate a student rechecking final answers before the final submission.

(iii) Assignment Features: These features seek to capture performance of student on graded assignments.

- **#A**: is the total number of assignments pertaining to one course. Assignments having a raw score of 0 possible_points were not counted as they bear no relevance to the final student grade.

- **ASubmission**: is the amount of assignment submissions for each student over a specific time duration. The intuition for choosing this feature is identical to QSubmission.

- **AScore**: is the total of assignment scores normalized by class average.

(iv) Request Features: Request features seek to capture student engagement.

- **AvgLoginHour**: is the average number of hours logged by a student per day over the entire course period. To filter useless requests we set the evaluate scale to one hour. As long as a student requests LMS in one hour that hour is considered to be a "Working Hour." The working hour rate and **CourseLen** can further display a student's engagement characteristics. The formula for this feature is given by:

$$AvgLoginHour = \frac{WorkingHour}{CourseLen} \tag{3}$$

- **AvgLoginDay**: captures the fraction of 24-hour cycles where the student has a request from the LMS over the entire course period. As long as a student make requests in a single day, we consider that day to be a "Working Day." The rate of the working day and **CourseLen** can demonstrate a student's engagement characteristics. We can view student engagement features from a various views using different evaluation scales (Hour, Day). The formula for this feature is shown below. The graphics representation of AvgLoginHour and AvgLoginDay for a sample student is shown in Figure 5.

$$AvgLoginDay = \frac{WorkingDay}{CourseLen} \tag{4}$$

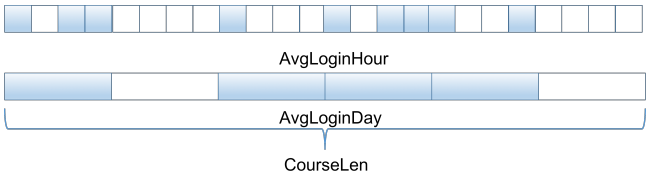


Figure 5: AvgLoginDay and AvgLoginHour and blue blocks are working periods

Final Student Performance

We discretized the final passing grades using a binary output. If the student's final grade was greater than the average score of the class (not counting students without scores) or the student's final grade was greater than 60, we assume this student to receive a passing grade for the course. We also consider **CourseLen**, **Type**, **Size**, **#Q** and **#A** as **Course features**. These are identical for every student within a class. The rest of the features are considered as **Student features** and are unique for every student and are associated wit a time-stamp parameter as described previously. We also combine the proposed student- and course-features in our study. The course features though unique for every student within a course differ as the student takes different courses and seek to capture patterns about the course that may correlate with student performance.

5 EXPERIMENTS

5.1 Datasets:

Discipline	# Courses
Business and Management	47
Computer Science	12
Education	85
Humanities	46
Interdisciplinary	22
Life Sciences	5
Mathematics,Statistics	18
Medical Pre-Medical	12
Other or Interdisciplinary	6
Physical Sciences	7
Professions and Applied Sciences	104
Social Sciences	12

Table 1: Course Principle Distribution for Canvas Courses

We performed empirical evaluation on dataset obtained from the Canvas Network Person-Course De-identified Open Dataset from 1/2014 to 9/2015 (https://dataverse.harvard.edu/dataverse/cn). This dataset consists of 376 courses across different disciplines listed in Table 1.

To evaluate our proposed approaches we sample the courses using the following three criterion: (i) Courses should have a start and end date because we can only identify the early stage of a course if the course has a distinct period. (ii) A suitable course should have a meaningful grade distribution. We filter out courses which do not report a final grade i.e., all the entries are '\N' or '0' in the final grade. (iii) The server logs have student request logs. Several of the courses within the Canvas dataset does not have any information pertaining to student enrolled within a course. This results in a total of 221 courses that satisfy all three conditions. These courses are referred by **Eligible-Courses** in the paper.

We performed two different types of experiments for the assessing the performance of early performance prediction within a class using early stage feature and described as follows:

5.2 Data Pre-processing:

For the Student-Specific approach, we sample students who have completed enough courses and with variance in their performance across the different courses. We choose students with a greater than a four course history and with at least two of the courses have passing and failing grades within the **Eligible-Course**. We found 586 students matching these criteria and their distribution is shown in Figure 7 with the **4363** student course performance data.

We used the stratified shuffle split method to achieve cross validation and the parameters used for this method list are displayed in Table 2. In the stratified shuffle split method, if we set the test set size to 0.1, this indicates that the testing set size is 10% of the overall dataset and remaining 90% for training data in each round. In each round, the training and testing dataset are randomly picked and this is no overlap between them. We set the test size to a smaller value because we want to set a prediction for a current student's course enrollment based upon their whole recorded course histories. A total of 436 testing data and 3927 training data records were used in this experiment.

Parameter	Description	Value
n_splits	Number of splitting iterations.	20/20
test_size	The size of the testing set.	0.1/0.1

Table 2: StratifiedShuffleSplit Parameters Table

For the Course-Specific approach, we choose courses from eligible course pools having more than 100 students with a final grade greater than 0. This selection resulted in 107 courses. The classroom size distribution is shown in the Figure 7. We use the stratified shuffle split method to split the training and testing data.

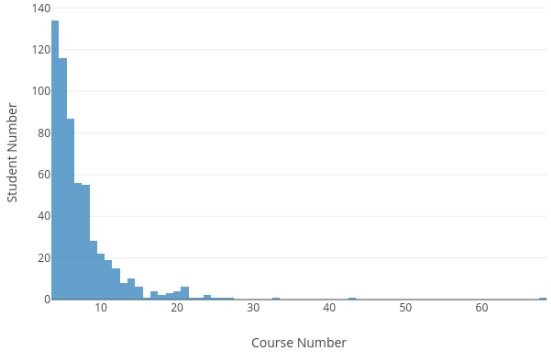


Figure 6: Course Number Distribution of 586 Students

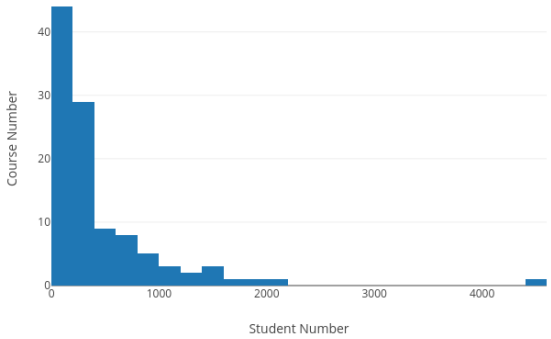


Figure 7: Student Number Distribution of 107 Courses

5.3 Evaluation Metrics:

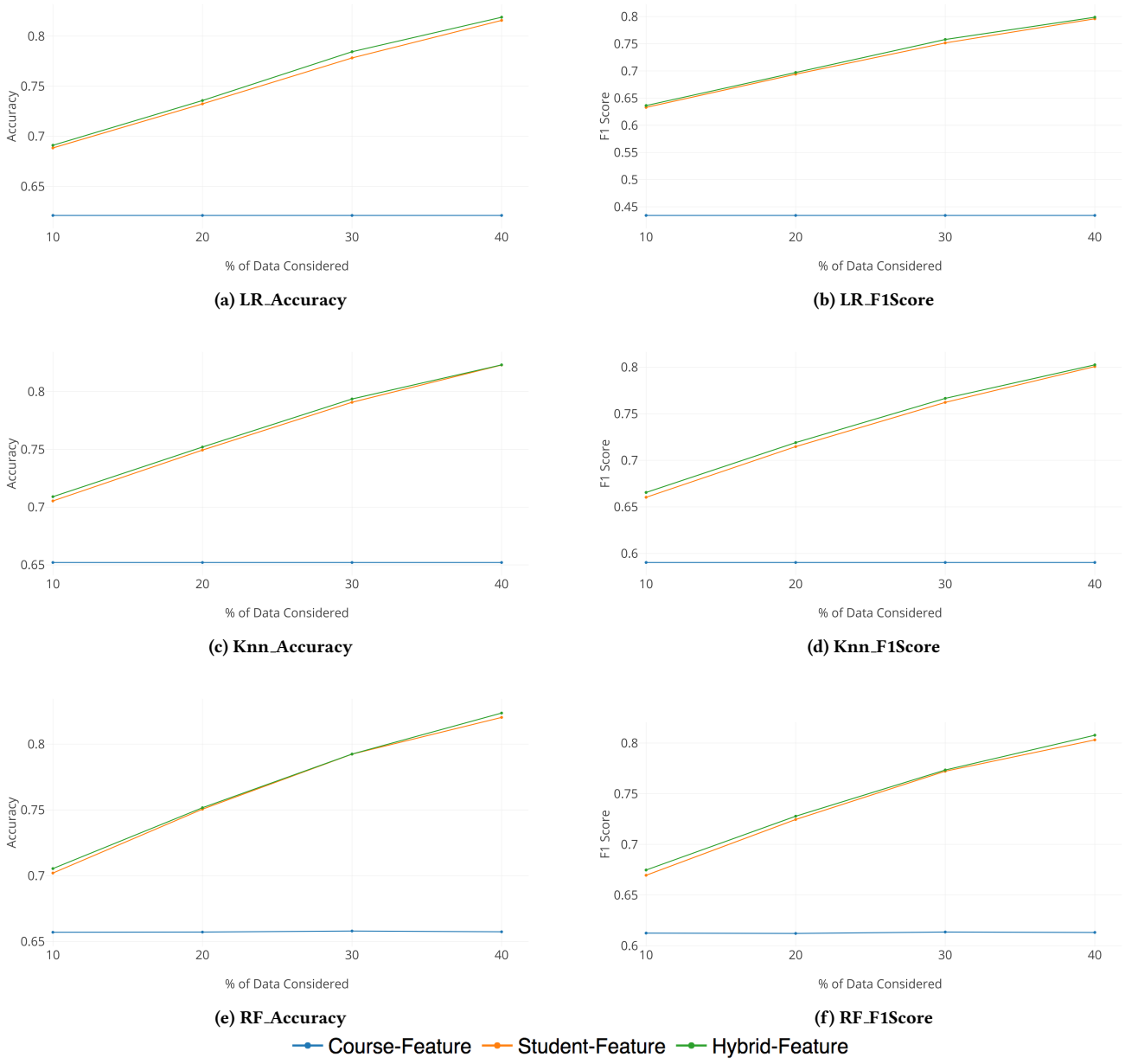
As we already normalize the grade output as binary and we can regard this prediction as a classification problem. For the classification problem, we use Precision and F1-score because it is a suitable metric for imbalanced datasets.

5.3.1 F1-score. *F1-score* is the harmonic mean between precision and recall. The higher F1-score means the precision (average accuracy) and recall score are higher than expected. F1-score's formula is given below with *TP* corresponding to true positive and *FN* corresponding to false negative.

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{5}$$

$$Precision = \frac{(TP)}{TP + FP} \tag{6}$$

$$Recall = \frac{(TP)}{TP + FN} \tag{7}$$



(a) LR_Accuracy **(b) LR_F1Score**

(c) Knn_Accuracy **(d) Knn_F1Score**

(e) RF_Accuracy **(f) RF_F1Score**

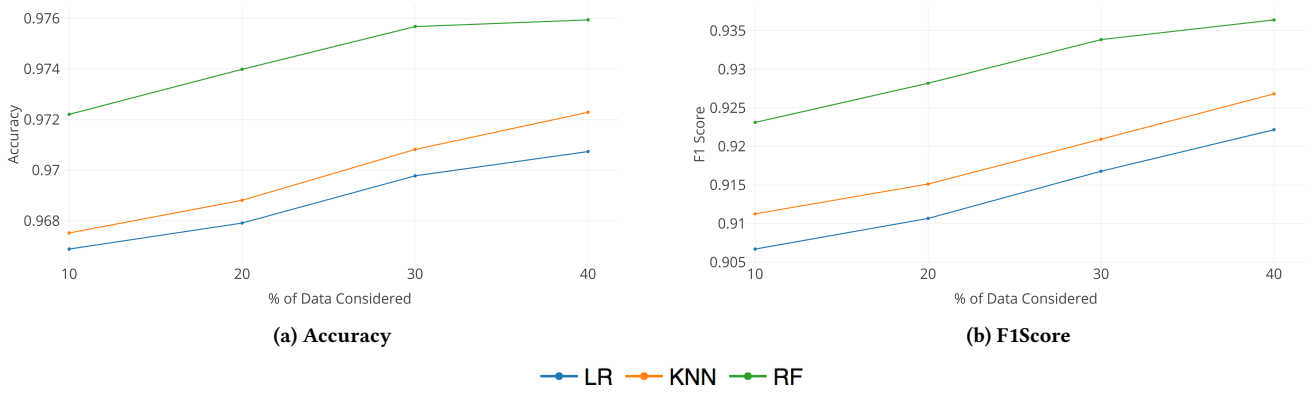
Figure 8: Average accuracy and F1 score using student, hybrid features respectively for three different classification method.

5.3.2 Gini Importance. Gini importance is the first commonly used importance measure method from RF [19]. RF derives a score for each feature to summary it's discriminative power. Gini index is used as a measure of impurity for a given attribute with respect to the output class. It is used to calculate the times a feature is used to split a node in RF [24]. The range of Gini importance is between 0 and 1.

6 RESULTS AND DISCUSSION:

6.1 Student-Specific Approach:

Figure 8 shows the accuracy and f1 scores for the classification performance varying the amount of data seen for training and testing. Results are reported for three different machine learning algorithms. The x-axis shows the percentage of data considered. For example, 10% denotes that we use the first 10% of the feature of training and testing data. We also evaluate the three types of



(a) Accuracy **(b) F1Score**

Figure 9: Average accuracy and F1 score for Course Specific Approach.

features created: (i) course-feature, (ii) student-feature and (iii) hybrid features.

The Figures show that the f1 score and accuracy increases from 70% to 80% as we increase the amount of data used for features from 10% to 40%. It is expected to determine the student's final grade and risk of dropping a class as the course draws to a close. The strong performance of the prediction methods early on shows the promise of making intervention decision early.

We also observe that Random Forest methods outperform the logistic regression and nearest neighbor algorithms. We also report accuracy results using both, the course and student features. These result mirror the experiment conducted by [6], which show far more accurate prediction methods by addition of features beyond grade-related within the prediction framework. In summary, given 40% of data uses (indicates course already passed 40%), we report 82.7% accuracy and 80.9% f1 score. Even for 10% of observed data, we report 71% accuracy and 67% f1 score. For the student-specific approach, the hybrid features within the random forest framework proves to be the winner.

6.2 Course-Specific Approach:

Figure 9 shows the performance of the methods with the Course-Specific Approach. We report good accuracy and F1 scores with the Course-Specific approach. Specifically, we observe approximately 95% accuracy and 90% f1 score using just the first 10% of the data/features. As we increase the amount of data used for making predictions (from 10% to 40%) for the final grade we do not see a substantial change. For a given course, the features related to interaction patterns computed for a given student are similar as time progresses within a semester. As such, there is little variance between early stage features and final stage features.

The Random Forest approach was the best performing method in the Course-Specific approach, followed by logistic regression and nearest neighbor algorithm. We noticed a similar trend for the Student-Specific approach as well.

As noted before this approach still has limitations. For now, we only conduct experiments in the same class with part of the students from this course are simulated as students enrolled in the particular course for the first time. In real-world scenario it is not guaranteed that two courses offered in different semesters would share the same feature types and can vary from semester to semester. Even though the overall course structure such as material covered and assignment types would not change much across different semesters, specific features like the professor/instructor, class size, and student interest/aptitude and habits will change. The Course-Specific Approach though shows better results than the Student-specific approach should account for the assumptions discussed above.

The Student-Specific approach only works well if a student has a longer course history (more courses are taken over a longer period). If we have enough have detailed enough students course performance records, we will consider the Student-Specific Approach to be the best predictor for the outcome.

6.3 Feature Importance:

We also evaluated the effect of each feature used in predicting the outcome and ranked these features for each early stage from 10% to 40% by using Gini index measure. Tables 3 and 4 show the feature importance of the Student-Specific approach and the Course-Specific approach, respectively. The higher value in the Table correlates with feature importance. Analyzing these results we observed that for the Student-Specific approach we notice that early during a semester the key positive feature is Ascore (Assignment Score). The course feature is not influential in our approach, but these course features were found to reduce the error in the early prediction to some extent.

For the Course-Specific approach since the course feature would remain static for each course here we only focus on the student engagement feature. We can see all the engagement features except QSubmission and QScore are positively correlated with the the final score.

7 CONCLUSION AND FUTURE WORK:

In this study, we developed a framework to predict the student's final class performance based upon features extracted from LMS and especially from the first few weeks of the semester. In our

Time-Stamp	10%	20%	30%	40%
CourseLen	0.066	0.054	0.05	0.045
Type	0.034	0.028	0.025	0.021
#A	0.047	0.04	0.034	0.031
#Q	0.046	0.039	0.032	0.029
Size	0.075	0.065	0.055	0.046
ASubmission	0.096	0.134	0.161	0.177
AScore	**0.147**	**0.162**	**0.203**	**0.211**
AvgLoginHour	0.142	0.14	0.11	0.094
AvgLoginDay	0.099	0.085	0.074	0.069
QSubmission	0.052	0.061	0.066	0.07
QScore	0.055	0.077	0.099	0.12
QAttempt	0.037	0.037	0.029	0.037
QTime	0.103	0.078	0.064	0.052

Table 3: Feature Importance for Student-Specific Approach

Time-Stamp	10%	20%	30%	40%
ASubmission	0.025	0.045	0.065	0.086
AScore	0.035	0.063	0.088	0.111
AvgLoginHour	0.037	0.038	0.037	0.04
AvgLoginDay	0.095	0.114	0.118	0.111
QSubmission	0.311	0.278	0.257	0.238
QScore	**0.393**	**0.356**	**0.33**	**0.308**
QAttempt	0.008	0.011	0.012	0.014
QTime	0.026	0.025	0.024	0.021

Table 4: Feature Importance for Course-Specific Approach

Student-Specific approach, even though the prediction based on the early feature did not perform as well as the complete features, the approach still achieved close to 83% accuracy over using 40% feature for current courses. Random forest approach was found to be the most suitable for the algorithm. The course features such as the course size and assignment counts do not affect the learning outcome but still reduce the error of prediction. We propose to consider non-grade related features such as gender, citizenship and professor/instructor in the future. Regardless we plan to enhance our approach to handle the student with the fewest records. In the Course-Specific approach, the machine learning methods achieved nearly 95% accuracy using the first 10% features.

Educational institutions attempt to identify some at-risk students by traditional methods such as mid-term reports. These are then used to communicate with students and setup meetings with advisors. However, it remains difficult for entire departments to monitor the performance of each student in every course, especially during the early stage. Our proposed approach allows for the prospect of real-time in-class performance prediction of students and can be later used for identification of at-risk students and provide them timely and much needed help.

8 ACKNOWLEDGEMENT:

This work is supported by Anonymous.

REFERENCES

[1] Mina Shirvani Boroujeni and Pierre Dillenbourg. 2018. Discovery and temporal analysis of latent study patterns in MOOC interaction sequences. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. ACM, 206–215.

[2] John P Campbell, Peter B DeBlois, and Diana G Oblinger. 2007. Academic analytics: A new tool for a new era. *EDUCAUSE review* 42, 4 (2007), 40.

[3] Tismy Devasia, TP Vinushree, and Vinayak Hegde. 2016. Prediction of students performance using Educational Data Mining. In *Data Mining and Advanced Computing (SAPIENCE), International Conference on*. IEEE, 91–95.

[4] Ramón Díaz-Uriarte and Sara Alvarez De Andres. 2006. Gene selection and classification of microarray data using random forest. *BMC bioinformatics* 7, 1 (2006), 3.

[5] Jiazhen He, James Bailey, Benjamin IP Rubinstein, and Rui Zhang. 2015. Identifying At-Risk Students in Massive Open Online Courses.. In *AAAI*. 1749–1755.

[6] Martin Hlosta, Zdenek Zdrahal, and Jaroslav Zendulka. 2017. Ouroboros: early identification of at-risk students without models based on legacy data. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*. ACM, 6–15.

[7] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied logistic regression*. Vol. 398. John Wiley & Sons.

[8] Ya-Han Hu, Chia-Lun Lo, and Sheng-Pao Shih. 2014. Developing early warning systems to predict studentsfi online learning performance. *Computers in Human Behavior* 36 (2014), 469–478.

[9] Suhang Jiang, Adrienne Williams, Katerina Schenke, Mark Warschauer, and Diane O'dowd. 2014. Predicting MOOC performance with week 1 behavior. In *Educational Data Mining 2014*.

[10] Anjeela Jokhan, Bibhya Sharma, and Shaveen Singh. 2018. Early warning system as a predictor for student performance in higher education blended courses. *Studies in Higher Education* (2018), 1–12.

[11] Sotiris Kotsiantis, Christos Pierrakeas, and Panagiotis Pintelas. 2004. PREDICTING STUDENTS'PERFORMANCE IN DISTANCE LEARNING USING MACHINE LEARNING TECHNIQUES. *Applied Artificial Intelligence* 18, 5 (2004), 411–426.

[12] Jean Lave, Etienne Wenger, and Etienne Wenger. 1991. *Situated learning: Legitimate peripheral participation*. Vol. 521423740. Cambridge university press Cambridge.

[13] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.

[14] Griet Lust, Jan Elen, and Geraldine Clarebout. 2013. Studentsfi tool-use within a web enhanced course: Explanatory mechanisms of studentsfi tool-use pattern. *Computers in Human Behavior* 29, 5 (2013).

[15] Joel McFarland, Bill Hussar, Cristobal de Brey, Tom Snyder, Xiaolei Wang, Sidney Wilkinson-Flicker, Semhar Gebrekristos, Jijun Zhang, Amy Rathbun, Amy Barmer, et al. 2017. The Condition of Education 2017. NCES 2017-144. *National Center for Education Statistics* (2017).

[16] Kew Si Na and Zaidatun Tasir. 2017. Identifying at-risk students in online learning by analysing learning behaviour: A systematic review. In *Big Data and Analytics (ICBDA), 2017 IEEE Conference on*. IEEE, 118–123.

[17] Canvas Network. 2016. Canvas Network Person-Course (1/2014 - 9/2015) De-Identified Open Dataset. https://doi.org/10.7910/DVN/1XORAL

[18] Kathleen Pittman. 2008. *Comparison of data mining techniques used to predict student retention*. Nova Southeastern University.

[19] Yanjun Qi. 2012. Random forest for bioinformatics. In *Ensemble machine learning*. Springer, 307–323.

[20] Zhiyun Ren, Huzefa Rangwala, and Aditya Johri. 2016. Predicting performance on MOOC assessments using multi-regression models. *arXiv preprint arXiv:1605.02269* (2016).

[21] Cristóbal Romero, Sebastián Ventura, Pedro G Espejo, and César Hervás. 2008. Data mining algorithms to classify students. In *Educational data mining 2008*.

[22] Mark Schneider and Lu Yin. 2011. The hidden costs of community colleges. *American Institutes for Research* (2011).

[23] Peter Brusilovsky Shaghayegh Sahebi. 2018. Student Performance Prediction by Discovering Inter-Activity Relations. In *Educational data mining 2018*.

[24] Archana Venkataraman, Thomas J Whitford, Carl-Fredrik Westin, Polina Golland, and Marek Kubicki. 2012. Whole brain resting state functional connectivity abnormalities in schizophrenia. *Schizophrenia research* 139, 1-3 (2012), 7–12.

[25] Li Zhang and Huzefa Rangwala. 2018. Early Identification of At-Risk Students Using Iterative Logistic Regression. In *International Conference on Artificial Intelligence in Education*. Springer, 613–626.