Learning Analytics for Computer Programming Education

Debabrata Mallick[1]

Student Dept. of Computer Science and Engineering, IIT Bombay, Maharashtra, India

---***---

Abstract - *Nowadays, in computer science, the student must have superb programming skill due to academic and professional life but these days, CS overall students having lack of this kind of quality. In this paper, they find an approach that they can track those students who are falling behind in computer courses and provide students with adaptive feedback that they can improve, and the teacher found those students who are not spending enough time and effort. They proposed a predictive model using several machine learning and deep learning techniques to train data. To train their predictive model, they use two kinds of data: student static and dynamic data. Before starting, of course, they collect student static data like (previous academic records, previous high school data, and student characteristics) based on course enrollment of students, and during the semester, every week, they collect student dynamic data like (activity log, amount spend of time in the assignment, online dynamic). These two types of data are first normalized and used for generating predictions based on their proposed model. From their prediction result, they send each student customized feedback via email. At the end of the semester, those students flow each week guidance they do well in the course.*

Key Words: Learning analytics; self-regulated learning; MOOCs; collaborative programming; big data analytics.

1. INTRODUCTION

Computer Science (CS) classes lack the coding performance of some students and need to improve(Lu et al., 2017). However, which students are lagging behind in CS classes, and finding them by the teacher is difficult when the class size is huge. Most of the time, the teacher will be able to find those students when the student fails in the exam. But if we are able to find those students before failing in the exam and we will be able to guide them, then it will be very beneficial for the student, and the failing chance will decrease. That is why we need automatic detection to predict those students. Nowadays, technology has evolved a lot because computer programming classes are conduct in blended ways; such as teachers teach the student in the classroom with face-to-face interaction and use several online tools like online assignment submission, self-assessment quizzes, Learning Management System (LMS). Nevertheless, all this helps students and teachers to learn and teach but using this online tool, we can collect data and leverage students digital footprint.

1.1 Multiple Data Sources

Data is everything in this research because the whole thing depends on data. Train the model or predicting the student at risk using advanced data mining techniques or providing adaptive feedback via email all depends on data. Furthermore, these different types of data are collected in several ways. Only one-time static data is collected before the start of the course, where dynamic data are collected each week. Both are difficult to collect, but dynamic collection data are relatively more complex.

1.2 Proposed Model

To identify those students who need assistance, we propose a prediction model using available data sources. For collecting, we student clickstream data, dynamic effort, and engagement. After collecting data, we leveraged the main component from the data and trained our proposed model using several advanced data mining techniques ten-fold. After training, calculate the result till the first week to semester mid.

1.3 Feedback

Some students might be opt-out after mid. That is why we start giving feedback to students those are still opt-in and predict the result. Based on the proposed predictive model result, provide custom massage feedback via email top to bottom or bottom to top. However, each student will get their addictive feedback, and the teacher will divide the resulting top student with the resulting bottom student in a group so that the lower result student will be able to learn from teamwork and improve their performance.

2. Digital Footprint

Student data are collected in several ways, also storing multiple times and in different locations. Like a student need to collect several locations and several ways. By collecting these data, we can leverage student engagement and involvement on the campus. All the way can collect Student data from multiple places, but we can evaluate a student based on their current academic performance at the university. A student might have excellent results in previous, but the performance in the university can be poor due to several reasons. If a student spends a fair amount of time in computer programming classes, that student has good chances to do well in assignments and examinations. During an assignment, a student spends much time in the module to complete the programming assignment and solve the problem. After solving the programming assignment student and taste the actual output or may submit again for a better result but all these time is pending and solving problem create a huge amount of digital footprint prints. All these things are handled by an online tool or platform, but these tools help students to submit and compiling their code. For teachers to evaluate the students submitted program or assignment, but there is no mandatory to use this only on the tool they can use several kinds of stools as their requirement. So for this problem, they collected data from several online educational platforms of student interaction. Therefore it is tough to collect all ground-truth data to train a predictive model. To solving this problem, they collect data from three different data sources for student engagement. Students' efforts in computer programming assignment or their engagement and interaction throughout the assignment or program and then build a model to achieve good predictive performance.

- At first, they calculate student characteristics and demographics from student registration.
- They build a custom learning platform for teaching computer programming for students to submit programs and instructors to review them.
- Finally, from the Learning Management System (LMS) system University collects student click-stream data for finding each student's engagement with the course material.

2.1 Context

This study research is done by Dublin City University in 2016(Azcona et al., 2019) . To helping teachers and identifying those is students are at risk to apply their proposed predictive model in their own University academic courses, they build a custom virtual learning environment for computer programming courses across computer science in Dublin City University. They design their learning management system where students and drag and drop their files, and the system will provide real-time feedback to the student. Even the system grades the student submission of assignments. The uses of student learning management system they collect student log data using artificial intelligence and machine learning techniques. Then they combine those data with training their model and finding the student is at risk in computer programming classes based on the result, which means predictive model results. Then provide their own adaptive customize feedback, but the feedback will provide after the mid-term exam because some student will opt-out before mid. The course was about twelve weeks, so after six weeks, they start guiding students about their problems and sometimes making group teams.

3. Predictive Modelling

Building the predictive model the primary purpose is to identify and classify those students have chances to fail in computer programming laboratory assignment. Using this predictive model, they try to find between lower performance and higher-performing students. Like a not equal number of students, the primary imbalance between the two classes is passing and failing in an exam. They collect student previous academic data before and during the course; they also collect student digital footprints to train their predictive model. From this predictive model, the used multiple machine learning technique used to extract information automatically using classification patterns and identifying the student performance in computer programming-based formal assignments. The result of the predictive model is quite dynamic and gives near to the expected result. They used the predictive model to predict the result on a timely basis like they predict whether students are diverting to their expected goal or not every week.

3.1 Data Processing

Data process using learning function for every week to identify among with higher and lower performing students best on their extracted data features for training the classifier. From the beginning to the end of the course, a set of dynamic and static data are used for each weekly learning function. Before the beginning of the course, all the static features are collected. Static features are student characteristics and students' previous academic history, or we can say their period of academic performance. And then, the dynamic features are collected every week during the course based on student engagement and progression, and interaction throughout the course and platform.

3.1.1 Feature Engineering

Every week, build a new classifier to predict, and each weak classifier predicts some prediction based on the classifier or learning function and model. Then from the predicted results, collect the top features and the student characteristics. Then, this top features and new week dynamic data are used for next week's prediction, and the processes are running throughout the whole course.

3.1.2 Static Feature

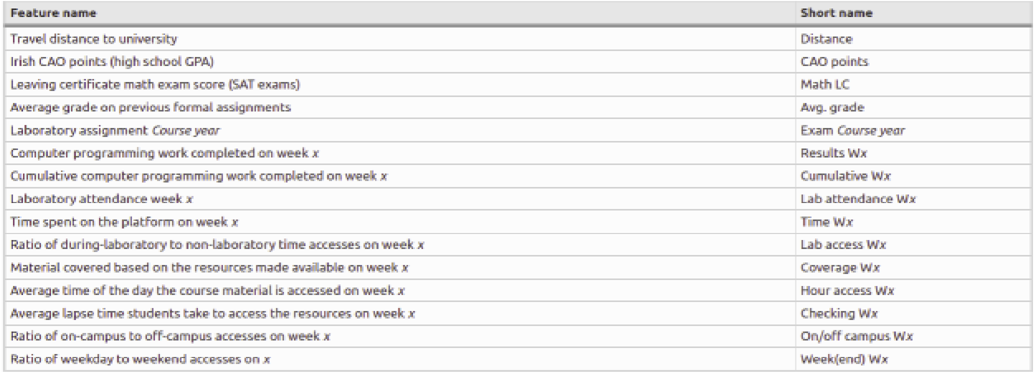
Static data collect before starting the courses, and this data is collected based on student characteristics and academic performance.

- Student characteristics: Their date of birth, distance from University to home, and how they come to University like the route to University.
- Prior academic performance: In this part, collect the previous academic results like CAO points, High school CGPA, and academic history in the previous university course.

3.1.3 Dynamic Feature

During the course is, dynamic data collected based on students' interactions throughout the week. Amount spent on time to access the course material and computer programming submission.

- Engagement: Get a system allowing them to submit their programming assignment in the system their submitted program and give real-time feedback, so it calculates the correctness of laboratory work and how much is completed.
- Prior academic performance: Try to find about how much time they spend on their laboratory work, during the course how they spend time on the platform how much time they used to access course material besides that they are accessing the course material based on IP address from home or campus and does they accessing the course material during the weekend.



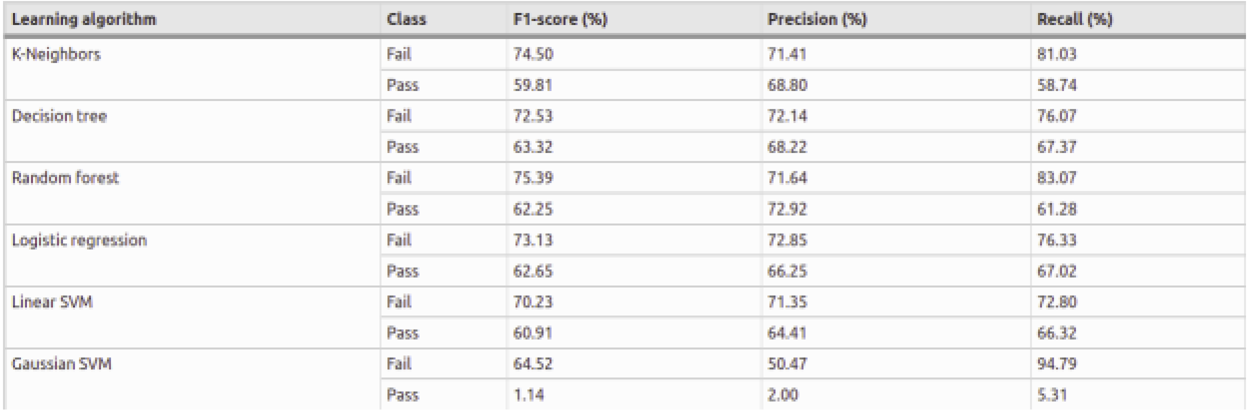
Feature name	Short name
Travel distance to university	Distance
Irish CAO points (high school GPA)	CAO points
Leaving certificate math exam score (SAT exams)	Math LC
Average grade on previous formal assignments	Avg. grade
Laboratory assignment *Course year*	Exam *Course year*
Computer programming work completed on week *x*	Results W*x*
Cumulative computer programming work completed on week *x*	Cumulative W*x*
Laboratory attendance week *x*	Lab attendance W*x*
Time spent on the platform on week *x*	Time W*x*
Ratio of during-laboratory to non-laboratory time accesses on week *x*	Lab access W*x*
Material covered based on the resources made available on week *x*	Coverage W*x*
Average time of the day the course material is accessed on week *x*	Hour access W*x*
Average lapse time students take to access the resources on week *x*	Checking W*x*
Ratio of on-campus to off-campus accesses on week *x*	On/off campus W*x*
Ratio of weekday to weekend accesses on *x*	Week(end) W*x*

Figure 3.1: Feature names and short names experimental setup from the paper.

3.2 Training the Model

For their predictive model, they use a classifier set to predict every week the chance of a student passing or failing in a computer programming laboratory exam. The course is divided into two parts mid-term and end-sem. They predict the student has the chance to fail in the mid-term exam on the top of six weeks and 7 to 12 week they predict the outcome of laboratory exam. Collected student static and dynamic data used to train every week learning function. The learning function builds by combining student dynamic and static data where student data is collected in previous but dynamic data features are from previous weeks. This process continued throughout the courses till the 12th week.

They use the empirical risks minimization principle to find out the misclassification error and find out which advance machine learning algorithm lowest empirical error in their learning function. Empirical risk minimization is easy for a family of learning algorithms that are helping to give the model with minimal average error over the training sets. They use a Scikit-learn python library to find which supervised and unsupervised algorithm provides a better solution. Their learning function or bag classifier contain several machine learning algorithm, and they want to find out which classifier performs well all above them because they want to pick the best classifier to which can accurately find out passing and failing in their prediction problem. However, finding this is quite difficult because some students might fail in the assignment but pass the exam. But their main objective is to find out those students are having issues during the course and solving the programming assignment. So, That is, we can guide them by providing feedback for grouping in a team.



Learning algorithm	Class	F1-score (%)	Precision (%)	Recall (%)
K-Neighbors	Fail	74.50	71.41	81.03
	Pass	59.81	68.80	58.74
Decision tree	Fail	72.53	72.14	76.07
	Pass	63.32	68.22	67.37
Random forest	Fail	75.39	71.64	83.07
	Pass	62.25	72.92	61.28
Logistic regression	Fail	73.13	72.85	76.33
	Pass	62.65	66.25	67.02
Linear SVM	Fail	70.23	71.35	72.80
	Pass	60.91	64.41	66.32
Gaussian SVM	Fail	64.52	50.47	94.79
	Pass	1.14	2.00	5.31

Figure 3.2: Performance of the learning algorithms in the bag of classifiers when trained and cross-validated from the original paper.

Their bag classified has a group of meta estimators that helps to find the subset of the original data set and aggregate with their predictions to find the final or best classifier.In the learning function, you have several advanced data mining algorithms like linear support vector machine (SVM), Gaussian SVM, Decision tree, Random forest, Logistic regression, and K-Neighbors. K-Neighbors, SVM, Random forest, Decision tree, Linear SVM this algorithm are used for both classification and regression, which helps to analyze and understand the relation between two or more variables. Where Logistic regression helps to categorize values in binary response of variable based on a mathematical equation and Gaussian SVM is it angle method used in SVM model.

Figure 3.2 describes the learning algorithm performance in the back classifier after training the data and cross-validation. The performance metrics of the learning algorithm show the average result of passing and failing in CS2 courses during the 12th week time period. Used Ten fold cross-validation to evaluate the predictive model by partitioning the original sample into a training set to train the model for every week's average result (Kohavi et al., 1995). To finding high-performance results in F1 matric day calculator precision and recall based on precision and recall value, they calculate F1 matric value. From learning algorithms, the F1 matric found that K-Neighbors has the highest chance of failing class on week 5 and 6.

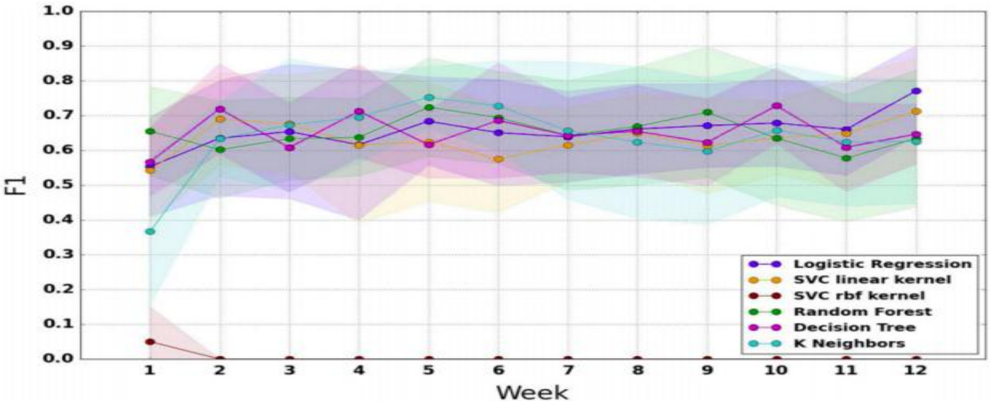


Figure 3.3: Performance of the bag of classifiers for the F1 metric from the original paper.

In figure 3.3 show that the result is good ever is the result of different force during the semester. The figure shows that only one machine learning algorithm learning function fails to detect.

Moreover, figure 3.4 describes that the learning function of the F1 matric value for the fail in the class.

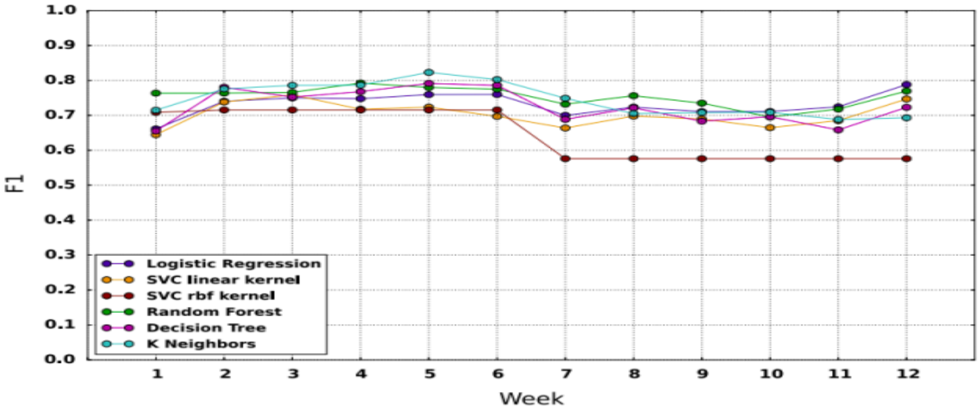


Figure 3.4: Performance of the bag of classifiers for the F1 metric and the fail class from the original paper.

The result shows that SVM with Gaussian kernel fails to detect students are passing means the expected P value of Gaussian is less than 0.00001. Besides only this SVM Gaussian algorithm, other learning algorithm functions on bag classifiers can predict, and they all have pretty similar results. If they had more data of a student, then they might be able to generalize better results.

4. Feedback

After generating predictive model results, they pick the most appropriate learning algorithms results and provide weekly personalized notification feedback or guidelines via email after the first literary exam in week six. Then release a feature in the module that a student can opt-in or opt-out for getting this notification or guidance notification via email. Once the student opt-in, they will not be able to change it during the course, but those students do not reply or opt-out, they will remove from the learning management system module.

Here is a list of personalized feedback and how they provide personalized feedback.

- For providing personalized feedback, they guide every student based on student performance on that week. Suppose a student's performance is between the top 10 percent. The system provides a customized massage with guidance.
- Same those students are below 10 percent get a customized message like please try to make more effort. In some other scenario, they also send eight custom messages between the students.
- Those students do not spend enough time on their programming assignments or courses, and the system will generate a message and send it in bold letters.
- Suppose a student does not participate in laboratory sessions or lab work. The system sends the message to the student to join the next lab station, and the tutor will resolve issues if there.
- If a student submits an assignment and the submitted assignment failed with the test case, then in the system generated message to solve the previous problem and if that specific student failed to solve previous problems, then the teacher makes the team with top students that the weak student can able to to learn before next week submission.
- Also, the system provides resources to learn and provides prediction guidance on how the system will predict or observe their weekly engagement throughout the course.
- Also, the student can unsubscribe from the getting notification via email, but nobody did that throughout the course.

5. Conclusion

Many students are currently studying in the computer science department worldwide, and they required super programming skills for doing well in the computer science field. Dublin City University found that computer science student lacks programming skills and the lack remains throughout the student University life. Furthermore, the class size is quite significant. That is why a teacher was not able to give proper guidelines and identify those students who are at risk. So that is why they propose a predictive model to identify the student are at risk in computer programming courses. To identifying those students who are at risk, they use student static and dynamic data to train their proposed predictive model using several advanced machine learning algorithms on their bag of the classifier to find those students who are at risk, from their predictive model using static student data and every week student dynamic data. They generate a weekly report based on that report and identify those students who are at risk. Those students are not concentrating or not focusing on the courses or even identifying those students are having some other problem. From the prediction model result, those students are at risk and give customized feedback via email. They also compare week reports and extract main features and use them to generate the following week report.

5.1 Feature Work

The proposed predictive model has just one-year student cohorts data, and the result came significantly well, but in the future, they can use several years of student cohorts data, which will help to have a better result. Moreover, adding some more modalities to the system like video recording during the lab session. So that they can track student behavior and identify students at risk, and based on that, they can provide a better guide to the students. Then the system could have a more advanced method like tokenism. In the future and they can remove opt-out students. This means those students who drop the course or opt-out after mid and are also getting the same kind of customized feedback messages and email.

REFERENCES

[1] Azcona, D., Hsiao, I.-H., and Smeaton, A. F., 2019, "Detecting students-at-risk in computer programming classes with learning analytics from students' digital footprints," User Modeling and User-Adapted Interaction 29, 759–788.

[2] Kohavi, R., et al., 1995, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Ijcai, Vol. 14 (Montreal, Canada). pp. 1137–1145.

[3] Lu, O. H., Huang, J. C., Huang, A. Y., and Yang, S. J., 2017, "Applying learning analytics for improving students engagement and learning outcomes in an moocs enabled collaborative programming course," Interactive Learning Environments 25, 220–234.

[4] Bravo, C., Duque, R., & Gallardo, J. (2013). A groupware system to support collaborative programming: Design and experiences. Journal of Systems and Software, 86(7), 1759–1771.

BIOGRAPHIES

Complected M.Tech CSE at IIT Bombay, Maharashtra.