A Case Study on Prediction of Student Performance in a Blended Learning Class when Using Small Data

SANCHEZ MCNEW BYRON ISRAEL†1 MITSUISHI TAKASHI†2
AOKI TERUMASA†3 ZHAO XIUMIN†4

Abstract: In this case study, we try to predict a student quiz score using a combination of the data obtained from a smart phone application used to study, and previous quiz information from the same students. We mainly focus on improving the identification of students at risk of failing the next quiz, in the scenario where the amount of data is limited, meaning, we do not have past data from other students, nor do we have many students to enhance our prediction accuracy. In order to improve our success rate in identification, we use an over-sampling and under-sampling method called SMOTE + Tomek Links, and also use a voting strategy we have called Safe Voting. In general, the use of these methods improves significantly the identification of students at risk of failing, though the accuracy of some classification algorithms is affected.

Keywords: Student Prediction, Learning Analytics, Machine Learning

1. Introduction *

The collection of educational data and information has increased greatly in recent years, and more and more educational institutions are adopting virtual learning systems and computer aided learning approaches in education. The problem with this is the fact that much of the data, although necessary to collect, is not used for anything rather than plain logging. Nonetheless, some institutions are starting to take advantage of the collected data and putting it to good use with learning analytics and data mining [1].

The case study of this research focuses on 1 part of a 3-phased learning process designed to create an educational learning system in a blended learning scenario that promotes continuous and sustainable learning. This educational environment includes a micro-learning environment and smartphone application as previously mentioned [2]. Both designed to help struggling students in class by providing teacher to student feedback, and after-class learning exercises created to help with the processing of new lessons and information. On a regular basis, students must attend class, share and learn from the teacher, and then proceed to practice the learned lesson at home using the application.

One of the biggest challenges with this approach is being able to identify students in need of assistance, and even more so, predicting which ones are prone to having problems in a near future. Therefore, we find ourselves in need of way of identifying students that are having academic performance problems (AP), this given the fact that the data collected for the class is all new data.

Most of the existing research on Educational Data Mining (EDM) focuses on how to predict future results based on historic data, which is available from the records of past courses [3]. However, in most scenarios, compiled and organized historic data is not available for analysis.

Another obstacle that we must tackle is that the amount of data available is not substantial, and therefore it becomes quite hard to predict with a high level of accuracy what student is in need of help and which one is not [4].

The method proposed in this case study uses the recently obtained data as historic data, in order to predict if a student needs academic assistance, in other words, if he will fail the next quiz, which can eventually lead to dropping out of a class. In addition to this, in order to make up for the small amount of data, it creates different features out of the existing data and different representations of it using an over and under sampling method (SMOTE + Tomek Links) [5], [6]. Additionally, a voting scheme we have called Safe Voting is also used in order to increment the recall value of the predictions.

The rest of the paper is organized the following way. Section 2 discusses related work, which served as basis for this research. Next, section 3 explains the overall functionality of the method proposed along with the ordered steps of execution. Section 4 presents the results of the experiment along with a description of the data used, as well as some of the problems encountered. Finally, section 5 includes the main conclusions of the paper and section 6 presents future works.

2. Related Work

2.1 Existing research

When we search for literature on learning analytics today, we are able to find a lot of research regarding prediction of student performance and drop out prediction, and general algorithms to perform data mining [7]. That is because in between the many objectives of learning analytics, we find that Knowledge Discovery in Databases (aka Data Mining) is one the most interesting and promising areas around. Data mining is defined as "the process of discovering useful patterns or knowledge from data" [8], and therefore has great interest in the field of education.

On this topic, Professor Ueno designed a learning system

* †1 Graduate School of Information Science, Tohoku University
†2 Institute for Excellence in Higher Education, Tohoku University
†3 New Industry Creation Hatchery Center, Tohoku University
†4 Center for Educational Informatics, Tohoku University

called Samurai, to help detect outliers using historic data and a Bayesian predictive distribution [9]. The research focused on finding students with irregular e-learning processes/patterns using prior knowledge of the response time characteristics of each content, and the learner's ability parameters. He would then proceed to use a decision tree in order to send motivating messages to students, given these patterns and the expect outcomes of them.

A paper on the changes and future of Data Mining in the educational context talks about the different classification of data mining and what research to focus on for each classification [3]. In general, data mining methods fall into the categories of supervised learning (or classification and prediction), unsupervised learning (or clustering), and association rule mining [10]. In addition, as is so, the use of classifiers has had a lot focus in recent years.

Pardo used recursive partitioning and a list of known available actions in an e-learning system to predict academic performance [11]. By analyzing a large number of numeric features obtained from the interaction with the system, the method automatically selected the most robust according to their performance. Ade applied an ensemble of different classifiers in order to further increment the accuracy of predictive models in a dataset of over 250 samples and 10 different attributes [12]. By combining the expected outcomes of two well know algorithms using different voting strategies, a combination of Naive Bayes and k-Star algorithms showed promising results with a 3% increment to the predictive accuracy level of the next highest performing algorithm it was compared to. In these researches, high predictive accuracy levels were achieved in their respective contexts using a large amount of historic data and features. In this paper by Jai Ruby, they comparted the predictive power of a single machine learning algorithm, using different subsets of data [13]. He compared the results of a MLP using both purely academic and academic, economic & personal information, when measuring student performance. Now although these papers are not key to this research, they do provide many of the tools we used such as the implementation of a voting scheme and feature creation, and the use of machine learning in the educational context.

Bote recently predicted whether the engagement of students in a MOOC would increase or decrease by analyzing their behavior and the actions performed in the system [14]. He used the data that became available during the course to create models for upcoming classwork, in other words, the data receive from the previous lesson would help predict the actions of the next. By creating different features focused on the actions performed on videos, exercises and assignments; using a CFS method for feature selection; and an SGD algorithm for classification, his team was able to detect disengagement of students at an early stage.

Hlosta also identified students at risk of failing a course using a model based on non-legacy data, but rather on data recently obtained in a VLE (Virtual Learning Environment) [15]. Ouroboros (the method's name) is a self-learning approach that uses the patterns from student who have just submitted an assignment in order to predict if other students will also submit. Both these researches address the topic of prediction using non-legacy data (recently collected information) and do so successfully using data obtained from educational tools, and therefore are key to this research. Yet still, they both point out a particular issue that is, a lot of data is necessary.

Many of the tools required to create predictions based on current data have been already introduced and proven successful in their respective contexts. The problem now is the lack of data available, or in other words, working with small data sets in educational context. Maharani generated new synthetic data using SMOTE for a small group of students [16]. They used a Naïve Bayes classifier in order to identify important characteristics in students that would determine their academic performance in class, given the answers of a questionnaire.

The uniqueness of the method proposed in this research is that it searches to address the issue of predicting student performance when no historic data and a small data set are available. We also try to bring together some of the many good ideas brought by each one of these researches, and provide a way of using this information to help students perform better in class.

2.2 Problems with existing research

The 2 most influential researches on this topic are the before mentioned works by Hlosta [15] and Bote [14], later mentioned as approach A (using data from students who have already handed in their assignments to predict for those who haven't) and approach B (using data from the past unit, to predict the next one), respectively. When performing these methods with the data that we have available from our scenario the results obtained are not satisfactory and do not meet the needs of our goal. Table 1 shows the accuracy results for both methods using different machine learning algorithms and our case study data.

Table 1: Past research testing

	Approach A - Accuracy	Approach B - Accuracy
MLP	0.51	0.48
RF	0.78	0.51
SVM	0.78	0.49

- Approach A has high accuracy values, by which it successfully predicted a large percentage of the students available.
- Approach B has accuracy scores ± 0.02 from the 0.50 marc, which indicates that only 50% of the students are being successfully predicted. In other words, Approach B is guessing in most cases.

In order to understand the high values for accuracy in Approach A, we performed a more thorough analysis to determine the Recall, Precision and F1-Score.

Table 2: Approach A analysis

	Precision	Recall	Accuracy	F1-Score
MLP	0.4	0.11	0.51	0.172549

RF	0.78	0	0.78	0
SVM	0.78	0	0.78	0

Table 2 shows the Precision-Recall results under Approach A. These results have high accuracy yet low general predictor performance (given by the F1-Score), due to the class imbalance problem [4].

3. Proposed Method

3.1 Method overview

The method is based on two main concepts: prediction using machine learning & recent data, and the expansion of small data. The use of current data and machine learning, serve the purpose of predicting the result of the current week, under the premise that student behavior gives the same academic results in the form of a pattern [17]. The expansion of small data serves the purpose of finding hidden relationships in the limited amount of data available and solving the class imbalance phenomenon [4]. By combining them, we seek to predict future student actions by finding hidden patterns in the expanded small amount of data that we have, and use these patterns to infer on a student's next action. Figure 1 shows a general workflow of the 10 steps involved in the process.

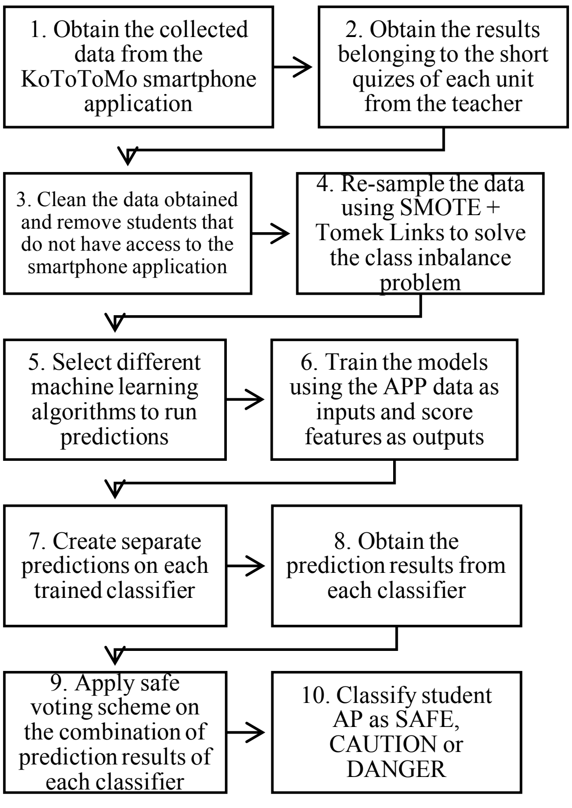

Figure 1: Method Workflow

3.2 Over and under-sampling

SMOTE [5] is an over-sampling technique in which instead of adding replacements to balance the classes, "synthetic" samples are created from the combination of the existing data from the actual limited samples. In SMOTE, the minority class is over sampled by taking each minority class instance and adding synthetic examples along the line segments joining any/all of the *k* minority class nearest neighbors. Depending upon the amount of over sampling required, neighbors from the *k* nearest neighbors are randomly chosen.

Tomek Links is the popular name for the pairs of minimally distanced nearest neighbors of opposite classes [6], in other words, instances that are so close alike, they could almost be part of the opposite class. As an under-sampling technique, the removal of these unwanted Tomek Links has been used in data processing and data mining. Because we want to use Tomek Links as an under-sampling method, we only eliminate instances of the majority group [18].

Since, both over-sampling and under sampling have their pros and cons, methods that combine both approaches have been developed, in order to balance the sampling process. One of the methods that perform both over and under-sampling is SMOTE + Tomek [19].

3.3 Voting Strategy and AP Classification

One of the major issues with predicting AP is that a student may change its study patterns quite easily, given that non-cognitive factors can promote academic success [20], and these are mostly factors we cannot completely control in a learning environment. Non-cognitive factors or skills, are a set of "attitudes, behaviors, and strategies" such as academic self-efficacy, self-control, motivation, expectancy and goal setting theories, emotional intelligence, and determination. Even so, there are educational models that seek to try to control these factors and have been successful doing so to some extent.

The reason why we mention this is that, in many cases, there is a thin line in between a student with good AP and a student with bad AP, and it is sometimes not as easy a task as classifying either good or bad, inclining or declining, fail or pass. Because of this, we define three simple but broad classifications for students' AP:

- SAFE: a student whose classification is SAFE, denotes that the likelihood of this student having decreasing AP is low
- CAUTION: a student whose classification is CAUTION, denotes that the likelihood of this student having decreasing AP is medium
- DANGER: a student whose classification is DANGER, denotes that the likelihood of this student having decreasing AP is high

These classifications have a conservative approach in which a student is always at risk of having decreasing AP, at different levels of probability. By adopting this approach, we seek to help as many students as possible, without disregarding any of them, yet prioritizing those in immediate need, or classified as DANGER.

In order to do so, we adopted a voting strategy we have called "Safe Voting". This approach works somewhat like a combination between veto voting strategy [21] and majority voting [22], where a single vote can veto the decision of whether the classification of a student's AP is SAFE, yet the majority rules to what extent. So, rather than voting on the classification of a student's AP, the voting is on the degree to which a student belongs to the group of declining AP. This differences from weighted voting [23] where, each classifier has a different weight towards the final decision. A somewhat similar approach has been seen in fuzzy clustering (or soft clustering) where the end result can belong to different classifications at the same time, but at different degrees [17]. We have adopted the concept of degrees and applied it to this context.

This voting strategy is based on the assumption that if a classifier predicts a student's AP to decline, this means that the student shares similar characteristics with other students who also have declining AP, and therefore has risk of declining AP. The overall strategy is to detect these students and tackle them by priorities, where a student in DANGER is high priority, a student in CAUTION is middle, and a student in SAFE is low. Table 3 shows the voting strategy.

Table 3: Safe Voting strategy

CLASSIFIER 1	CLASSIFIER 2	RESULT	VALUE
FAIL	FAIL	DANGER	0
FAIL	PASS	CAUTION	0
PASS	FAIL	CAUTION	0
PASS	PASS	SAFE	1

3.4 KoToToMo Application data

The KoToToMo smart phone application consists of four types of exercises, described in Table 4.

Table 4: KoToToMo Application Exercises

Type of exercise	Description
Repeating	The student is required to repeat out loud the sentences after listening to a video. This exercise consists of 1 task.
Speech recognition	The student is required to listen to an audio clip and recognize the spoken sentences. This exercise consists of 4 tasks.
Sentence pattern	The student is required to order the Chinese characters available to complete the sentence. This exercise consists of 6 tasks.
Shadowing	The student is required to mimic the speech with just a few seconds of delay as to speak together with the video. This exercise consists of 1 task.

Each exercise has its data recorded into an application database in a central server and is then recollected for analysis. The lessons range from number 1 to number 12 with increasingly difficulty and should be attempted as homework assignment for after the class is given. Students may attempt an exercise an unlimited amount of times, and only the highest score is recorded as the final score. The data available for each exercise is described in Table 5.

Table 5: KoToToMo Application data description

Data	Value	Example
CLASS ID	ID (1 to 7)	1
USER ID	Unique student ID	15
SUBMIT	Date of submission of entire exercise	6/18/2017 10:44:42 PM
STYLE OF STUDY	Categorical (type of exercise)	Sentence pattern
UNIT	The number on the unit studied	3
QUESTION	Sub question ID of the exercise type (only sentence pattern and strength test are applicable)	2
DURATION	Numeric seconds until submit	20
RESULT	right or wrong (only applicable to sentence pattern and speech recognition)	True (1)

3.5 Classifiers

We have performed our predictions using seven different classifiers with the purpose of covering different ways of handling the available data when doing prediction. The reason behind this is that we are somewhat unfamiliar with the behavior of the application data, and therefore wish to cover different approaches. We selected these algorithms for their variability in classification procedures and good performance in different scenarios. It is not the purpose of this research to explain machine learning, so therefore we will not give a detailed description of each method, but we have included enough detailed to understand its benefits and usage. Table 6 contains the list of classifiers used for prediction.

Table 6: Classifiers used

Classifier	Applications & Characteristics
Multilayer Perceptron	They can approximate virtually any function to a desired accuracy; however, results are only valid if there is a sufficiently large number of training samples.
Random Forest	They correct the problem of DTs of overfitting the data and try to reduce the variance, but at the expense of an increase in bias.
Support Vector Machine	One of the most efficient machine learning algorithms, mostly used for linear and non-linear patterns recognition. However, SVM lacks the ability to show scoring as parametric function and therefore is not very transparent.
Gaussian Naïve Bayes	It often competes with more sophisticated classifiers, and works well when features are completely independent, or sometimes with functional depended features. However, high entropy feature input space has low performance.
ADA Boost	It is fast, simple and easy to program, with no parameters to tune, and requires no prior

	knowledge of the weak hypothesis. It focuses on hard cases of classification and often finds outlier due to the weights assigned to each example. The drawbacks are that it is dependent on the data and may perform poorly if not enough data is available.
Gradient Boosting	It is specifically good at creating prediction procedures when using non-clean data, but has the same drawbacks of boosting methods such as being susceptible to noise and high dependency of data.
Decision Tree	One of the best advantages of this method is the ability to combine it with other classifications techniques, as it is simple to understand. The major disadvantages are that calculations can get complex, and information gain is biased in favor of attributes with more levels.

3.6 Feature selection methods

Various features representing the obtained application data are created from it. Table 7 shows these features.

Given these features, there are 4 methods for selecting the appropriate ones to use:

- Approach 1: Using PCA as a feature reducing method (down to 10 features)
- Approach 2: Using the top 10 features selected by a Pearson Correlation Coefficient [24]
- Approach 3: Using all the available features, with no reduction
- Approach 4: Doing individual predictions of each type of exercise and unifying the results using majority voting (PEMV)

Table 7: Features

Feature Name	Description
ATTEMPTS	How many attempts the student performed on this exercise for this unit
COMPLETED	How many attempts were completed (all tasks performed)
RESULT AVERAGE	The average result obtained per task
SCORE	Highest obtained result for an exercise
DIFFERENCE BETWEEN FIRST AND LAST ATTEMPT	The amount of days between the first and last attempt at an exercise
AVERAGE DURATION	Average in seconds for each attempt at the exercise
AVERAGE DAYS BEFORE QUIZ	Average of how many days before the quiz is the exercise performed the most
AVERAGE DAYS AFTER RECENT CLASS	Average of how many days after the most recent class is the application used the most
MORNING PRACTICE	If the average practice is performed during the period of time between 5:00 and 12:00
AFTERNOON PRACTICE	If the average practice is performed during the period of time between 12:00 and 19:00
EVENING PRACTICE	If the average practice is performed during the period of time between 19:00 and 5:00 of the next day
WEEKEND PRACTICE	If the average practice is performed during the weekend

4. Experimental Results

4.1 Data

In this study, we ran our proposed method with the data belonging to Japanese-Chinese language course. 7 classes with a total of 268 students, and 4 units were used for this case study. A student may attempt the same exercise many times without limit on the mobile application. Students are required by the teacher to use the application and have a score penalization for not using it at all. The score penalization is not considered for the prediction, but does serve as motivation and to explain behavior patterns.

4.2 Experimental part 1 – Sampling and Voting

The first section of the experiment will test the performance of the classifier algorithms with and without using sampling and the voting strategy under both approaches A and B.

First of we compare the use of sampling under both approaches

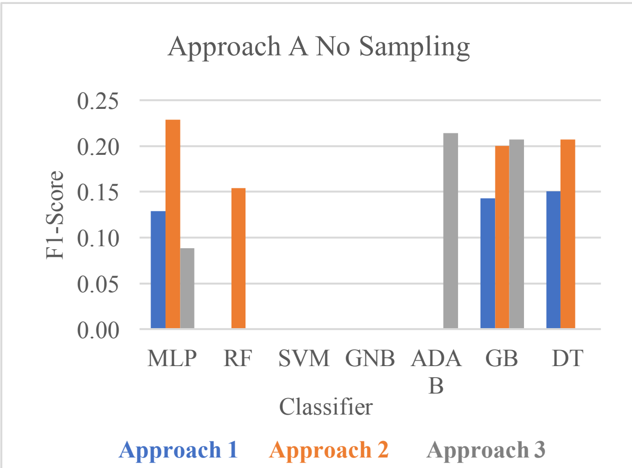

Figure 2: Approach A No Sampling

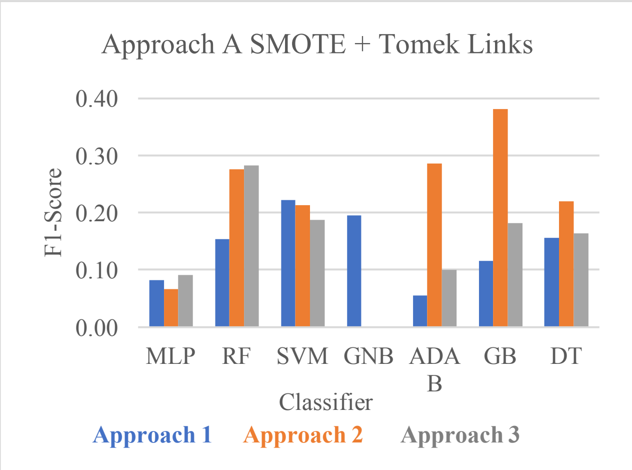

Figure 3: Approach A with Sampling

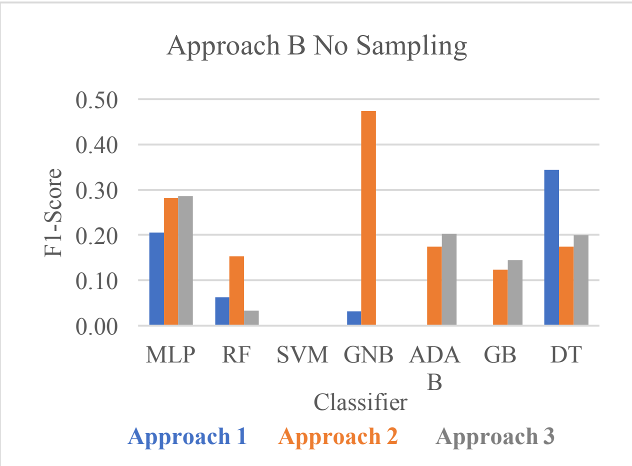

Figure 4: Approach B No Sampling

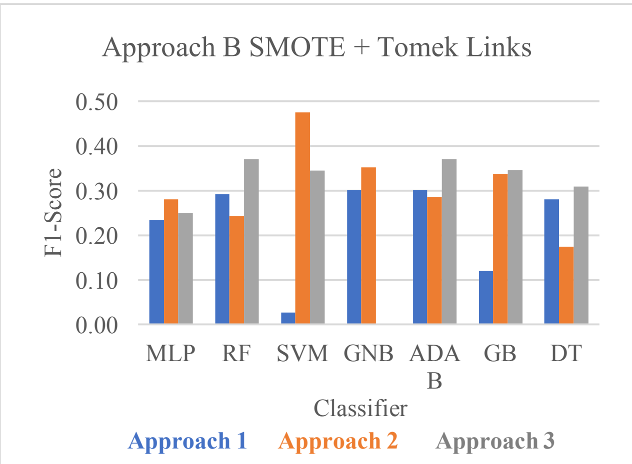

Figure 5: Approach B with Sampling

Given these results, we determine that the use of sampling does not consistently improve the results under Approach A, but has an overall positive impact on the results under Approach B. We therefore discard the use of Approach A for further testing.

We now compare the use of a voting strategy with Approaches 1 through 3 for Approach B.

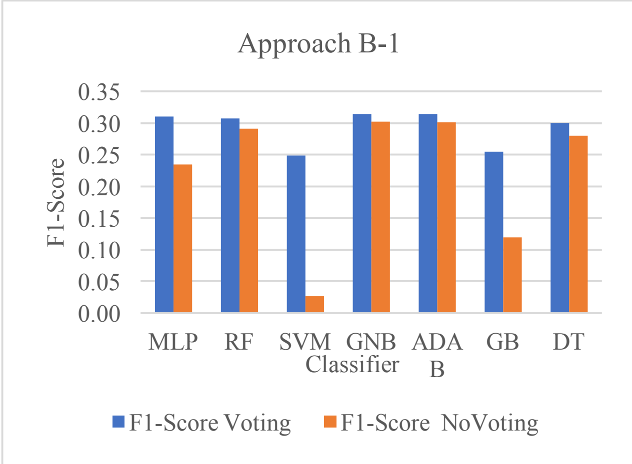

Figure 6: Approach B1

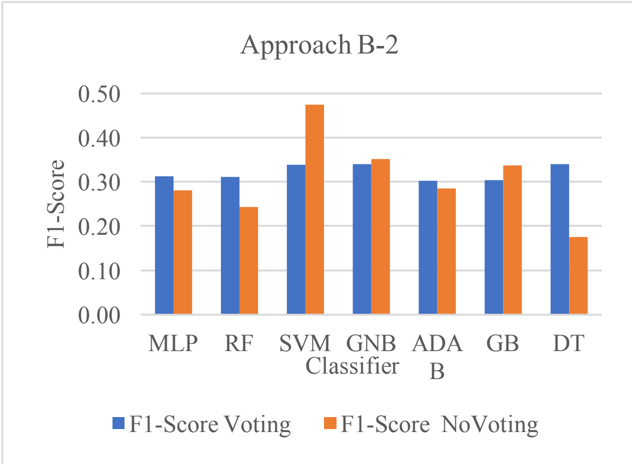

Figure 7: Approach B2

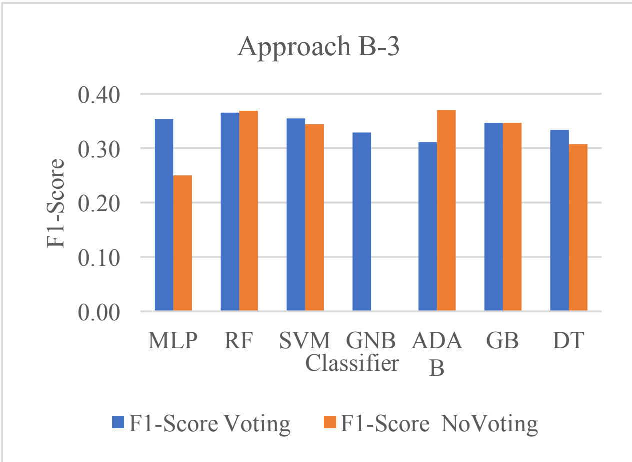

Figure 8: Approach B3

Given these results, we determined that the voting strategy improved the performance of the classifiers in 85% of the cases where it is applied. We therefore conclude that the use of a Smote + Tomek links and Safe Voting improves prediction performance under Approach B.

4.3 Experimental part 2 – Features and Classifiers

The second part of the experimentation seeks to determine the best combination of classifiers to use for the voting strategy, and the best feature selection method.

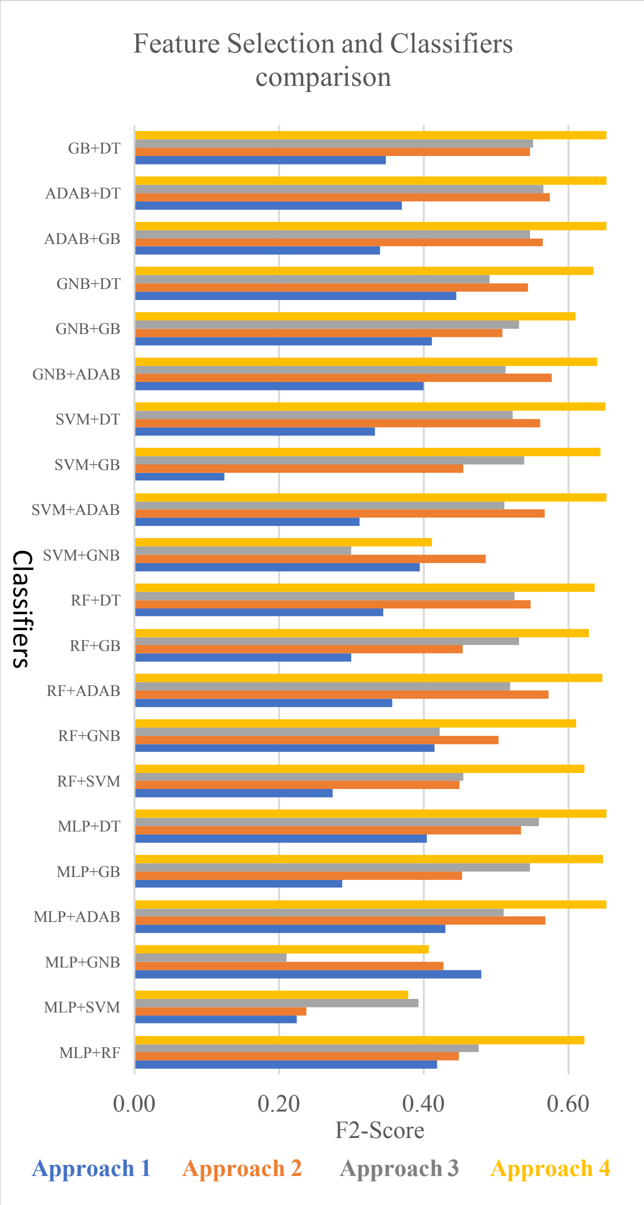

Figure 9: Feature Selection and Classifiers comparison

Given these results, we determined that the best feature selection method is Approach 4, and the best combination of features is GB + DT with an F2-Score of 0.67.

4.4 Student prioritization and error

Using the obtained prediction results we determined the classification of students given their performance and the resulting error. For this study case, we set a threshold of 70% of the score for the next short quiz, were anything below this is considered failed (declining AP). Figure 10 shows the ground truth of students who passed and failed. Figure 11 shows our predictions with the resulting error.

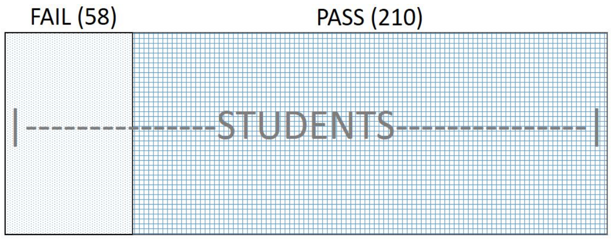

Figure 10: Ground truth of predictions

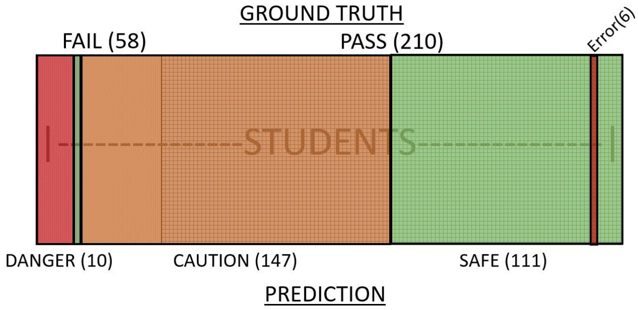

Figure 11: Predicted results and error

Using the obtained prediction results we determined the classification of students given their performance and the resulting error shows in Table 8.

Table 8: Prediction results and error

	Predicted Results	Expected Results
Students that Failed	58	58
Students that Passed	210	210
Students in DANGER	10	16
Students in CAUTION	147	147
Students in SAFE	110	104
Misclassified students	6	0

5. Conclusions

Given the results found by the experimentation of this case study, we are able to conclude that the use of the here proposed method enables the detection of student in need of assistance by using a combination of an over-sampling method (SMOTE) and an under-sampling method (removal of Tomek Links). In addition, the use of the voting strategy "Safe Voting" helps improve the accuracy of detection of these students in need of assistance. We also determined that the best feature selection method for this case study is using individual exercises for prediction. Overall, the goal of "*detecting students in need of assistance when a small amount of data is available*" was achieved.

6. Future Work

As future work for this case study, we have three principal points:

- Analyzing the existing error when predicting.

- Analyze the reason behind why combinations of classifiers GB +DT works bests with the data available for this case study.
- Try different over and under-sampling methods to identify the best combinations to work with the data available in this case study.

Reference

[1] C. Romero and S. Ventura, "Educational data mining: a review of the state of the art," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 40, no. 6, pp. 601–618, 2010.

[2] X. Zhao and L. Zhang, *KOTOTOMOーことばを友に—マルチメディア中国語初級テキスト*, 1st ed. 2017.

[3] P. Thakar, A. Mehta, and Manisha, "Performance analysis and prediction in educational data mining: A research travelogue," *Int. J. Comput. Appl.*, vol. 110, no. 15, pp. 60–68, 2015.

[4] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," *Proc. 2000 Int. Conf. Artif. Intell.*, pp. 111--117, 2000.

[5] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[6] T. Ivan, "Two Modifications of CNN," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-6, no. 11, pp. 769–772, 1976.

[7] J.-L. Hung and K. Zhang, "Revealing online learning behaviors and activity patterns and making predictions with data mining techniques in online teaching," *MERLOT J. Online Learn. Teach.*, vol. 4, no. 4, pp. 426–437, 2008.

[8] B. Liu, *Web Data Mining*. 2011.

[9] M. Ueno, "Online outlier detection of learners' irregular learning processes," *Data Min. E-Learning*, vol. 2, no. 1999, pp. 261–277, 2006.

[10] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2006.

[11] A. Pardo, N. Mirriahi, R. Martinez-Maldonado, J. Jovanovic, S. Dawson, and D. Gašević, "Generating actionable predictive models of academic performance," in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16*, 2016, pp. 474–478.

[12] A. Roshani and P. R. Deshmukh, "Classification of students by using an incremental ensemble of classifiers," in *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, 2014, pp. 1–5.

[13] J. Ruby and K. David, "Analysis of Influencing Factors in Predicting Students Performance Using MLP -A Comparative Study," *Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ.*, vol. 3297, no. 2, pp. 1085–1092, 2007.

[14] M. L. Bote-Lorenzo and E. Gómez-Sánchez, "Predicting the decrease of engagement indicators in a MOOC," in *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, 2017, pp. 143–147.

[15] M. Hlosta, Z. Zdrahal, and J. Zendulka, "Ouroboros," in *Proceedings of the Seventh International Learning Analytics & Knowledge Conference on - LAK '17*, 2017, pp. 6–15.

[16] K. M. Hidayah, T. B. Adji;, N. I. Akhmad, and S. Hidayah, "Comparison analysis of data mining methodology and student performance improvement influence factors in small data set," *Int. Conf. Sci. Inf. Technol.*, pp. 169–174, 2015.

[17] T. Levin, Z. Libman, and R. Amiad, "Behavioral patterns of students under an individualized learning strategy," *Instr. Sci.*, vol. 9, no. 1, pp. 85–100, 1980.

[18] E. T and A. M, "Classification of Imbalance Data using Tomek Link(T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method," *J. Informatics Data Min.*, vol. 1, no. 2, 2016.

[19] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM SIGKDD Explor. Newsl. - Spec. issue Learn. from imbalanced datasets*, vol. 6, no. 1, pp. 20–29, 2004.

[20] L. M. Gutman and I. Schoon, *The impact of non-cognitive skills on outcomes for young people: Literature review*, no. November. 2013.

[21] R. K. (Blekinge I. of T. Shahzad and N. (Blekinge I. of T. Lavesson, "Comparative Analysis of Voting Schemes for Ensemble-based Malware Detection," *Proc. 7th Int. Conf. Availability, Reliab. Secur.*, no. August, pp. 98–117, 2012.

[22] Y. Zhang, H. Zhang, J. Cai, and B. Yang, "A weighted voting classifier based on differential evolution," *Abstr. Appl. Anal.*, vol. 2014, 2014.

[23] L. I. Kuncheva and J. J. Rodríguez, "A weighted voting framework for classifiers ensembles," *Knowl. Inf. Syst.*, vol. 38, no. 2, pp. 259–275, 2014.

[24] J. Lee Rodgers and W. Alan Nice Wander, "Thirteen ways to look at the correlation coefficient," *Am. Stat.*, vol. 42, no. 1, pp. 59–66, 1988.

Acknowledgments This work was supported by the JSPS KAKENHI grant, numbers JP15K02709 and JP15K01012. I would also like to thank the members of both Aoki Lab and Mitsuishi Lab at Tohoku University for their insight and input into the topic.