symmetry

MDPI

Article

An Ensemble Prediction Model for Potential Student Recommendation Using Machine Learning

Lijuan Yan [1,2,*] and Yanshen Liu [1,2]

[1] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China; yanshenliu@mail.ccnu.edu.cn

[2] Hubei Research Center for Educational Informationization, Central China Normal University, Wuhan 430079, China

* Correspondence: yanlijuan@mails.ccnu.edu.cn

Received: 29 March 2020; Accepted: 20 April 2020; Published: 3 May 2020

Abstract: Student performance prediction has become a hot research topic. Most of the existing prediction models are built by a machine learning method. They are interested in prediction accuracy but pay less attention to interpretability. We propose a stacking ensemble model to predict and analyze student performance in academic competition. In this model, student performance is classified into two symmetrical categorical classes. To improve accuracy, three machine learning algorithms, including support vector machine (SVM), random forest, and AdaBoost are established in the first level and then integrated by logistic regression via stacking. A feature importance analysis was applied to identify important variables. The experimental data were collected from four academic years in Hankou University. According to comparative studies on five evaluation metrics (precision, recall, F1, error, and area under the receiver operating characteristic curve (AUC) in this analysis, the proposed model generally performs better than compared models. The important variables identified from the analysis are interpretable, they can be used as guidance to select potential students.

Keywords: ensemble; prediction model; student performance; machine learning

1. Introduction

To attend academic competition is an effective way to test teaching and learning performance, and it has a positive impact on students' scholarly motivation and study habits in education [1]. Academic competition is a series of activities to find and solve problems through practical activities outside the classrooms, which is an effective measure to identify and train young talents [2–5].

So, attending academic competitions is very beneficial to students. The academic competitions can improve students' study effectiveness during study activities, and enhance the students' collective efficacy as well as their awareness of collaboration and communication.

In China, application-oriented universities attach great importance to academic competition. Tutors are assigned to train students before the competition. However, there are two difficulties during the competition organization. Students lose the chance to win the prize due to not taking part in the competition. Tutors find it hard to select potential students since the number of students is too large. To achieve good results in the competition, it is very important to select potential students. Tutor observation or practice tests can select students who have outperforming performance. These methods are biased by the tutors' knowledge and experiences. Moreover, they are not suitable if there are too many students involved in the selection.

In this study, a prediction model is proposed to predict and analyze student performance in academic competition. By applying data mining to the students' background data and behavior data, the model predicts the competition results and then identifies key features, which affect prediction

results. This work has practical reference values to tutors on conducting and encouraging the potential students to attend the academic competitions.

2. Related Work

In recent years, it has become more and more popular to apply the machine learning methods to student performance prediction in different educational scenarios, and data sources used in the prediction have covered many subjects. A method was proposed to predict "At-risk Learners" [6–8] or end-of-learning performance [9] through analytics of online learning behavior data. Some researchers analyzed data in the course management system or student demographic information data to identify the factors that affect academic performance, and the research studies revealed the main factors include language family [10], sleep habit [11], computer usage [12,13], and so on. Other research focused on behavioral data collected in the classroom, which were used to analyze the correlations between test scores and different courses, and predicted a pass probability of the specific subjects [14,15] or degree qualified probability. Based on these analysis results, tutors can take interventions accordingly to optimize student learning efficiency. In terms of the construction of the academic performance prediction model, many researchers tend to adopt traditional machine learning methods, such as logistic regression, decision tree, artificial neural network (ANN), and support vector machine (SVM). These methods have been confirmed to be able to improve the prediction performance according to current studies. For example, Kotsiantis et al. [16] applied six algorithms (C4.5, back propagation, naive Bayes, 3-nearest neighbor, logistic regression, and sequential minimal optimization) to train the data set to predict poor performers, they found that the naive Bayes algorithm had better performance on satisfying accuracy. Romero et al. [17] compared the performance of different machine learning methods (decision trees, fuzzy rule induction, and neural networks) for predicting final marks of students. In the study, they applied discretization and rebalance preprocessing techniques to get better classifier models.

Ensemble learning [18] is the process by which multiple models (like classifiers or experts) are strategically generated and integrated to solve a computational intelligence problem. It uses multiple learning algorithms to get a better prediction performance than the performance obtained from a single algorithm. It has been applied to a wide range of topics in classification, regression, feature selection, and abnormal point detection. For instance, Beemer et al. [19] proposed an ensemble learning approach to estimate individualized treatment effects (ITE) to characterize at-risk students and to assess student success and retention under intervention strategies. The work by Ade Roshani and Deshmukh P. R. [20] applied an incremental ensemble consisting of three classifiers (naive Bayes, K-star, and SVM) and used a voting scheme to predict the career of students. Kotsiantis et al. [21] applied the ensemble methods to predict student success in distance learning with three different techniques (WINNOW, naive Bayes, and 1-nearest neighbor). These studies have confirmed that the ensemble model is more likely to attain higher accuracy than the single algorithm.

In summary, at present many studies have been done to predict students' academic performance in traditional classrooms or online learning platforms. These studies presented very interesting and reasonable results. The study on the prediction in the academic competition has been of concern to researchers.

3. Contribution and Paper Structure

A lot of relevant information can be collected in an academic competition, such as a student's demographic information, behavioral information (generated by students' daily study, and participation performance in competitions). It is almost impossible for a tutor to discover whether they have a connection with a student by personal experience alone when faced with a sizeable amount of data. Therefore, machine learning is a practical way to solve this problem.

In this study, we collect and construct a data set about the students' performance in academic competition. The data set contains all the information regarding participated students and their competition results (winning or losing the competition). Learning in a supervised fashion was used to discover structures in the data set, and the machine learning methods were applied to train a model

that can explain the data. At the same time, we used this model to predict new data, and to identify the key features that affect students' performance through feature importance analysis. The analysis results can be provided to tutors or managers, to be used as references for candidate selection.

Let $\{x_1, x_2, \cdots x_n\}$ represent the set of n characteristic variables for a student and y represents the student's competition results. The y has two possible values, winning or losing the competition, which is encoded as 1 and 0, respectively. So given the perspective of machine learning, candidate selection for student performance prediction in academic competitions can be formulated as a binary classification problem by grouping students into two symmetrical classes. The relations between characteristic variables and the target variable can be described as Equation (1).

$$y = f(x_1, x_2, \cdots, x_n) + \varepsilon. \quad (1)$$

where f represents an unknown function, which is the prediction model we intend to train using the collected data, ε is the error between the true target variable y's value for each student and the predicted value from the prediction model. The goal of the binary classification problem is to train a model that can predict the output of the target variable y when given a series of input variable x. The prediction y_{pre} of a new student x_{new} can be attained by $y_{pre} = f(x_{new})$.

In this study, a novel ensemble model based on stacking is proposed to predict student performance in academic competitions. This study tries to answer two questions:

- How can we improve the model prediction performance with an ensemble strategy?
- The predictive results need to be understandable. So how can we identify significant features? What suggestions can be made for tutors/managers based on the analysis results?

In this paper, Section 4 introduces the proposed ensemble model in detail. In Section 5, we describe the detailed analysis flow of the proposed ensemble learning model for a real data set. The validity and robustness of the model were verified, and the feature importance analysis was conducted by the model. The experimental results are presented and discussed in Section 5. In Section 6, we conclude and present a brief outlook for future studies.

4. Method

Different learning algorithms yield different results on the regression or classification problems. As the learning results from different algorithms are different, it is possible to improve the final prediction performance for each algorithm, so that better results can be obtained with the combined learning algorithms when compared to the results obtained by using a single algorithm [22]. Ensemble learning is designed to boost predictive performance by blending the predictions of multiple algorithms.

Ensemble learning has been commonly used in machine learning on a variety of classification and regression tasks to improve performance by grouping individual algorithms. Several types of ensemble methods, such as voting, averaging, bagging, boosting, and stacking, have been proposed. Stacking is an efficient heterogeneous ensemble method. It has been widely used in data mining competitions in recent years. It can be regarded as a super multi-layer perceptron. Each layer includes one or multiple models, and the next layer learns from the results of the previous layer of the model. In machine learning, many models are used to solve a binary classification problem, including regression algorithm, decision tree algorithm, kernel-based algorithm, Bayesian method algorithm, clustering algorithm, etc. The stacking can be integrated easily with different classifiers or regression models to improve robustness and generalization in a single model.

4.1. The Proposed Ensemble Model

In this study, we constructed an ensemble model using 2-layer stacking. The learning algorithms used in the first layer are called the Base-learner, and the algorithm in the second layer is called the Meta-learner respectively. The Meta-learner is used to combine the prediction results from all Base-learners.

The construction procedures of the prediction model can be described by the steps below:

Step 1. Divide the data set into training sets and test sets;

Step 2. Construct a new data set based on the output of Base-learners;

Step 3. Train the Meta-learner to output the final prediction result based on the newly constructed data set.

The 2-layer stacking model can be used to combine machine learning algorithms to boost predictive accuracy. When constructing the stacking ensemble model, the selection of the Base-learner and Meta-learner affects model performance.

The premise of using stacking to improve the classification effect is that the Base-learner should have a good prediction performance. In general, stacking will work best if the algorithms being combined are in some sense very different from one another. So, the bigger differences in the classification principle between these Base-learners, the more complementary they can be in the process of an ensemble. The classification results will be optimized accordingly. These strategies were applied to select the Base-learners in the first layer. Random forest (RF) [23–25], SVM [26–28], and AdaBoost [29,30] classifiers were chosen in this study to be the Base-learners due to their better classification performances. They are commonly adopted as predictive models for student achievement prediction at present and have good performance. The RF and AdaBoost algorithms were used for classification since they are both excellent ensemble algorithms based on decision trees, whereas RF produces multiple decision trees based on a randomly selected subset of training samples and variables. RF [27] does not need to assume a data distribution and can handle thousands of input variables without variable deletion. However, the tree structure of RF is unstable and may overfit training data, and therefore its generalization performance is poor. The SVM is a powerful and flexible machine learning model and can perform linear or nonlinear classification. The SVM is particularly well suited for the classification of complex data with small- or medium-size data sets. Although linear SVM classifiers are efficient and perform well in many applications, it works only for the linearly separable data and it is very sensitive to outliers. The AdaBoost emphasizes adaptivity through frequently modifying the sample weights and adding weak classifiers to boosting. The AdaBoost is sensitive to noisy data and outliers, and it is less susceptible to the overfitting problem than most of the other learning algorithms. These three models have their advantages and disadvantages when they are applied to the prediction of student performance. Appendix A of this paper provides detailed elaborations of the SVM, RF, and AdaBoost algorithms. The Base-learner in the stacking used a well-performed model in pursuit of adequate learning during data training. Therefore, the stacking is more prone to overfit the training data. To reduce the risk of overfitting, the Meta-learner tends to select simple models, such as logistic regression, lasso regression, and so on. We chose logistic regression as the Meta-learner in our study. Logistic regression is the basic model for the prediction of a dichotomous dependent random variable. Logistic regression does not have high general accuracy and is prone to underfit the training data. In addition, it is not suitable to deal with nonlinear features [31]. However, logistic regression is a good choice for the prediction of success in a course or program [32]. Appendix A of this paper provides a detailed elaboration of the logistic regression algorithms.

Finally, the SVM, RF, and AdaBoost classifiers were used as the Base-learners, and logistic regression was adopted as the Meta-learner in this study. An ensemble model using stacking was constructed as illustrated in Figure 1.

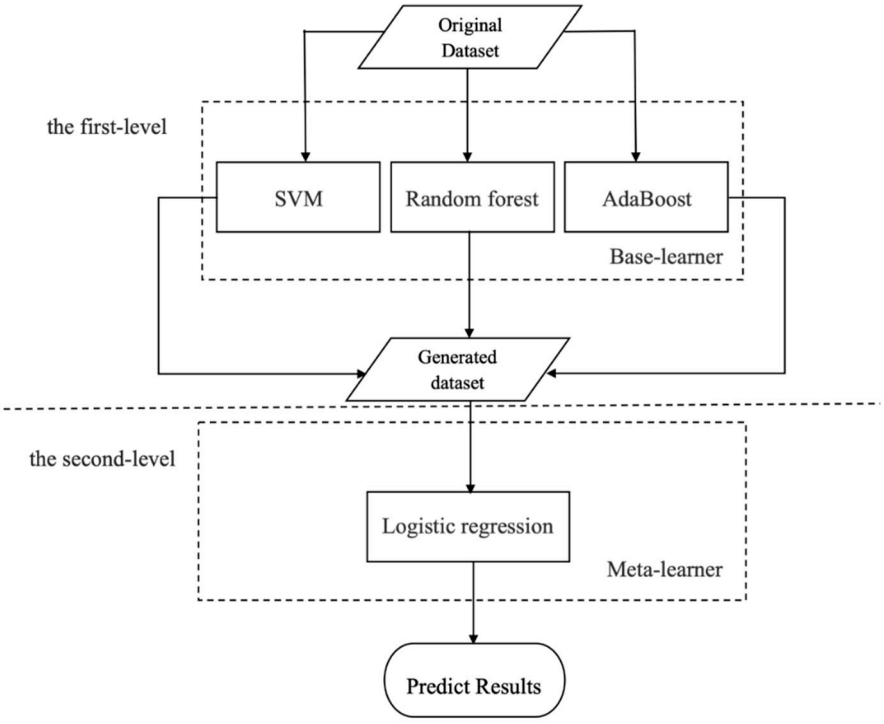


Figure 1. The proposed ensemble prediction model.

The characteristic variable x represents the student data, and the data are classified into three major types of information. The first type is the student's background information, the second type is the student's behavior data during the competition, and the last type is the student's daily academic performance. These raw variables are then divided into two types, one is numerical and the other one is categorical, as listed in Table 1.

Table 1. Description of raw variables. In the Type column, the N denotes a numerical variable and C denotes a categorical variable.

Category	Variable	Type	Description
Background	gender	C	Male, Female
	source_l	C	Place of residence of participants in the college entrance examination
	speciality	C	Computer science, Law, English, E-commerce, etc.
Behavior	enrollment_c	C	Subject category: Science, Literature, History, Engineering, etc.
	competition_g	C	Grade at the time of the first participation in the subject competition. Freshman, sophomore, junior, senior
	competition_n	N	Total number of participants in academic competitions
	peer_b	C	Whether a classmate of this major has participated in the same competition in the past? Yes or No
	competition_t	C	Whether to participate in the same competition multiple times? Yes or No
GPA(Grade Point Average)	pre_c_GPA	N	GPA for all semesters before participation in the competition
	c_GPA	N	GPA for the semester of competition
Result	result_c	C	Whether to win in the academic competition

At first, each Base-learner provides a probability that a sample belongs to each class, $P(y = 1|x)$ or $P(y = 0|x)$. Each classification model has two attribute prediction probabilities, so six prediction probabilities are obtained in total for each student sample. Secondly, the predicted attribute probability values from the Base-learner are provided for the Meta-learner as its input. The actual class label y will be used to learn and then get the final prediction model M. In the end, the final class label of prediction results will be given by the $\widetilde{M}$.

4.2. 10 × 10-Fold Cross-Validation

In 10-fold cross-validation, the data set is randomly divided into 10 equal parts. In turn, nine of them are taken as the training set and the other one is used as the testing set. The mean value of predictive results from the 10 turns is regarded as the evaluation result of the model [33].

While training a Meta-learner, the test results of the training set need to be characterized by learners from the previous layer. If we train the learner and then predict it in a training set, this will cause a label leak. A label leak means that the training models and prediction results display the personal information of participants that are contained in the data set. A label leak should be avoided when the stacking is applied. To avoid label leak, we use another 10-fold cross-validation in each training set. As shown in Figure 2, each Base-learner uses 10-fold cross-validation to generate a new feature in this study.

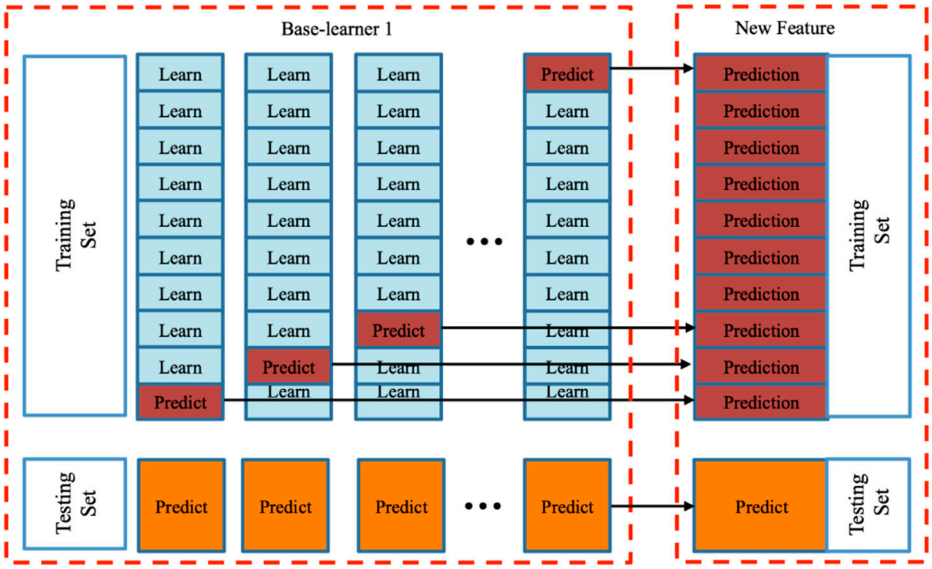


Figure 2. The 10×10-fold cross-validation model used as a Base-learner in this study.

Given the initial sample set is $\{(y_n, x_n), n = 1,2,\cdots,N\}$, where y_n is a class label $(y_n \epsilon \{0,1\})$ and x_n represents the characteristic variable. The sample-set is randomly divided into 10 subsets with the same size and the subsets are defined as $\{S_1, S_2, \cdots S_{10}\}$. We chose one of these subsets as a testing set each time, and the remaining subsets were regarded as training sets. We used the 10-fold cross-validation as shown in Figure 2 for each original training set. Taking the kth Base-learner as an example, each round of training and testing will produce two possible results, $P_k(y = 0|x_i)$ and $P_k(y = 1|x_i)$. Here the $P_k(y = 0|x_i)$ and $P_k(y = 1|x_i)$ are given by the kth Base-learner, and they represent the probabilities of the prediction result of the ith sample belonging to the class with label y = 0 (losing in competition) and $y = 1$ (winning in competition), respectively. Once the prediction results are obtained from the three Base-learners, all the probabilities will be concatenated together to form a new feature vector.

As shown in Figure 3, the feature vector is expressed in the form of a $N_{tr} \times 6$ matrix, represented by L in this study. In the matrix, N_{tr} is the number of samples in the training set, and each Base-learner contributes two dimensions.

$$\begin{bmatrix} P_1(y=0|x_1) & P_1(y=1|x_1) & P_2(y=0|x_1) & P_2(y=1|x_1) & P_3(y=0|x_1) & P_3(y=1|x_1) \\ P_1(y=0|x_2) & P_1(y=1|x_2) & P_2(y=0|x_2) & P_2(y=1|x_2) & P_3(y=0|x_2) & P_3(y=1|x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_1(y=0|x_{N_{tr}}) & P_1(y=1|x_{N_{tr}}) & P_2(y=0|x_{N_{tr}}) & P_2(y=1|x_{N_{tr}}) & P_3(y=0|x_{N_{tr}}) & P_3(y=1|x_{N_{tr}}) \end{bmatrix}$$

Figure 3. The feature vector in the form of an N tr×6 matrix.

5. Experiment Studies

5.1. Data Collection

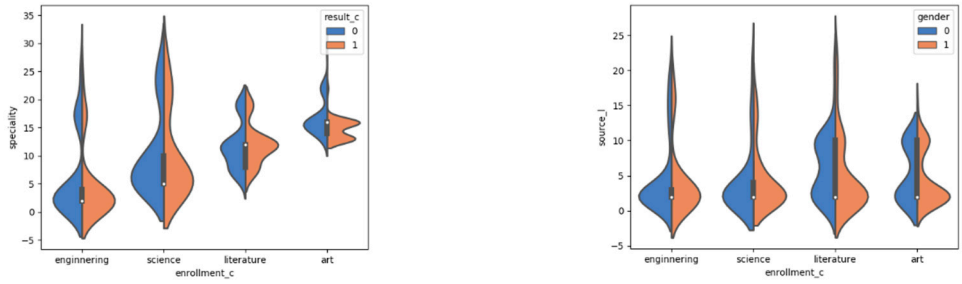
Data were collected from Hankou University, and included the records of undergraduate students who had participated in at least one national academic competition from 2015 to 2018 academic year. In total, 684 competition records containing 486 students are present in the data. There are 86 academic competitions covering multiple categories, such as engineering, literature, science, art, and so on. In addition, there are some comprehensive competitions included, such as innovation and entrepreneurship competition, skill competition, etc. We regard the 1st prize, the 2nd prize, and the 3rd prize awarded in the competitions as winning in this study.

For each student, eleven variables were defined, as described in Table 1. Ten of them are characteristic variables, and they are derived from the demographic information and the behavioral information of students. The above information was gathered from students' daily study performance and behavior in participating competitions. These variables were taken as inputs of the prediction model. The result of the competition was defined as the target variable. Data masking was applied to the data set to mask private or sensitive information before analysis.

5.2. Data Preprocessing and Description Analysis

Data preprocessing was conducted in the following steps: At first, seven of these raw variables were categorical variables, including gender, specialty, subject category, etc. They were converted to the numerical values by integer encoding. For example, Freshman, sophomore, junior, and senior are set to 1, 2, 3, and 4, respectively. Secondly, frequent itemsets were used to substitute the missing values.

The density distributions of specialty and enrollment_c were grouped according to the results of the competition as shown in Figure 4a. The density distributions of source_l and enrollment_c were grouped according to the gender as shown in Figure 4b. Generally, the enrollment_c and gender of students are evenly distributed, and engineering students are prone to participate in competitions.



(**a**) (**b**)

Figure 4. (**a**) Density distributions of specialty and enrollment_c grouped according to the results of the competition; (**b**) density distributions of source_l and enrollment_c grouped according to gender.

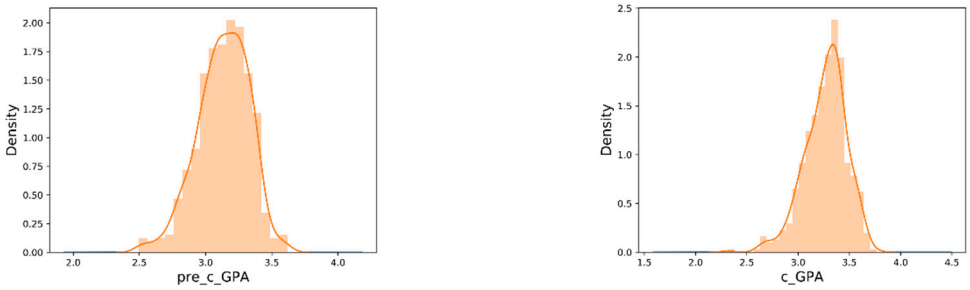
Table 2 shows the descriptive analysis results of the preprocessed four academic years' data. Skewness was used to calculate the skew direction and degree of statistical data distribution, and it is a digital feature of the degree of asymmetry of statistical data distribution. As shown in Table 2, most students chose to take part in academic competitions during sophomore and junior years. Different students have a different number of participations as shown in Table 2. For example, a student may participate in more than one academic competition.

Table 2. Descriptive statistics of the preprocessed data.

Variable	mean	Std dev.	min.	25%	50%	75%	max.
gender	0.45	0.50	0.00	0.00	0.00	1.00	1.00

competition_g	2.24	0.79	1.00	2.00	2.00	3.00	4.00
competition_n	1.60	1.28	1.00	1.00	1.00	2.00	11.00
peer_b	0.76	0.43	0.00	1.00	1.00	1.00	1.00
competition_t	0.13	0.34	0.00	0.00	0.00	0.00	1.00
pre_c_GPA	3.13	0.19	2.50	3.01	3.15	3.27	3.62
c_GPA	3.27	0.21	2.31	3.14	3.29	3.40	3.77
result_c	0.36	0.55	0.00	0.00	0.00	1.00	1.00

The frequency distributions of the pre_c_GPA and c_GPA are shown in Figure 5a,b, respectively. In short, both the pre_c_GPA and c_GPA are described well by the normal distribution. In total, 159 of 486 participated students won in the national academic competitions.



(**a**) (**b**)

Figure 5. (**a**) The frequency distribution of pre_c_GPA; (**b**) the frequency distribution of c_GPA.

5.3. Classification Performance Indicators

The models considered in this study were evaluated by five performance measures: $precision$, $recall$, $F1$, error, and $area\ under\ the\ receiver\ operating\ characteristic\ curve$ (AUC). The outputs of classifiers are summarized in four groups: The students who won the competition were correctly labeled as winner (TP); the students who lost the competition were incorrectly labeled as winner (FP); the students who lost the competition are correctly labeled as loser (TN); the students who won the competition are incorrectly labeled as loser (FN).

The *Precision* and *Recall* are calculated by Equations (2) and (3) as below:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Equation (2) indicates the precision metric as the fraction of students who won the competition over the students predicted as the winner. Indeed, the larger number of *FP*, the lower precision of the classifier. Moreover, to investigate the ability of the classification model in predicting all winners, we used the recall metric as defined in Equation (3). In other words, the recall metric represents the fraction of the students who are correctly labeled as a winner over the whole number of winners. It is worth mentioning that more students being labeled as winner will lead to high recall and low precision. The harmonic average of precision and recall, called $F1$, is also considered in this study and is described as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)\ (4)$$

The *F1* estimates the quality of classification for both winner and loser, simultaneously. Equation (5) below defines the measure *Error*, which means the proportion of the students wrongly labeled as winner and the students wrongly labeled as losers over all participated students.

$$Error = \frac{FP + FN}{TP + TN + FP + FN} \quad (5)$$

A receiver operating characteristic (ROC) curve is a graphical approach for analyzing the performance of a classifier. It uses a pair of statistics (true positive rate and false positive rate) to characterize a classification algorithm's performance. The resulting plot can be used to compare the relative performance of different classifiers and to determine whether a classifier performs better than random guessing. The ROC curve is not affected by the changes of the sample distribution. The AUC value represents the area under the ROC curve. The larger the AUC value is, the better the classification algorithm is. The AUC is equivalent to the probability of the case that a randomly selected positive example is ranked higher than a randomly selected negative example [34].

5.4. Results

In this study, an ensemble model using stacking was developed using the Python language within the Sklearn library framework in PyCharm. PyCharm is a python integrated development environment with a set of software that can help users improve their efficiency when developing in Python. The ensemble model was constructed using the RF, SVM, and AdaBoost as its three Base-learners, and logistic regression was selected as the Meta-learner. To test the effectiveness and stability of the proposed ensemble model, 10 prediction experiments were carried out. After performing the 10 rounds of the proposed ensemble model, the prediction results were obtained, as listed in Table 3. The average value of the 10 rounds was calculated as the overall prediction performance of the proposed ensemble model. As shown in Table 3, the $Precision$, $Recall$, $F1$, $Error$, and AUC are 0.8550, 0.8600, 0.85, 0.1460, and 0.9185, respectively.

The AUC values are classified in three performance levels with following thresholds: $AUC > 0.9$ (excellent), $0.7 < AUC < 0.9$ (fair), and $AUC < 0.7$ (poor). The results shown in Table 3 indicate that the proposed ensemble model has a better prediction performance.

Table 3. Test results of the proposed ensemble model from 10 runs.

Run index	Class label	Precision	Recall	F1	Error	AUC
1	0	0.8400	0.8800	0.8600	0.1429	0.9202
	1	0.8700	0.8400	0.8500		
2	0	0.8400	0.8700	0.8600	0.1480	0.9206
	1	0.8600	0.8400	0.8500		
3	0	0.8500	0.8700	0.8600	0.1429	0.9157
	1	0.8600	0.8600	0.8500		
4	0	0.8400	0.8600	0.8500	0.1531	0.9191
	1	0.8500	0.8400	0.8400		
5	0	0.8400	0.8700	0.8600	0.1480	0.9199
	1	0.8600	0.8400	0.8500		
6	0	0.8400	0.8800	0.8600	0.1429	0.9159
	1	0.8700	0.8400	0.8500		
7	0	0.8500	0.8700	0.8600	0.1480	0.9192
	1	0.8600	0.8500	0.8500		
8	0	0.8500	0.8900	0.8700	0.1378	0.9160
	1	0.8800	0.8400	0.8600		
9	0	0.8500	0.8800	0.8600	0.1378	0.9196
	1	0.8700	0.8500	0.8600		
10	0	0.8500	0.8800	0.8700	0.1378	0.9187
	1	0.8700	0.8500	0.8600		
Min		0.8400	0.8400	0.8400	0.1378	0.9157
Max		0.8800	0.8900	0.8700	0.1531	0.9206
Ave		0.8550	0.8600	0.8565	0.1439	0.9185

To get further clarity of the prediction performance for the proposed model, a comparative study was carried out, in which several algorithms were selected for performance comparison. These selected single algorithms are commonly used in the existing studies for student performance

prediction, including SVM, decision tree, logistic regression, and Bernoulli naive Bayes (BernoulliNB). The experimental results of the comparative study are summarized in Table 4.

Table 4. Classification results of compared with single algorithms.

Method	Class label	Precision	Recall	F1	Error	AUC
SVM	0	0.7757	0.8384	0.8058	0.2041	0.8444
	1	0.8202	0.7526	0.7849		
Decision Tree	0	0.7826	0.7273	0.7539	0.2395	0.7605
	1	0.7404	0.7938	0.7662		
Logistic regression	0	0.7419	0.6970	0.7188	0.2275	0.7725
	1	0.7087	0.7526	0.7300		
BernoulliNB	0	0.7500	0.6667	0.7059	0.2351	0.7649
	1	0.6944	0.7732	0.7317		
Proposed model	0	**0.8447**	**0.8788**	**0.8614**	**0.1429**	**0.9138**
	1	**0.8710**	**0.8351**	**0.8526**		

The results reveal that the prediction performance of the proposed ensemble model in this study is better than the other four models with a single algorithm each. The prediction results in this study are used for competition candidate selection, therefore, the model focuses on the merits of the indicators labeled as 1. The *Precision* of the proposed model gets the highest value of 0.8710, which is followed by the SVM model. Using the proposed model, the probability of students being correctly labeled as 1 is 87.1%. The value of *Recall* in the proposed model is 0.8351, which is followed by the decision tree. Among the five models, the proposed model has the best performance with the AUC value of 0.9138. At the same time, the proposed model has the lowest error rate compared to other algorithms.

According to the comparison of ROC curves, as shown in Figure 6, the areas under the curve (AUC) of all the five models are much larger than 0.5.

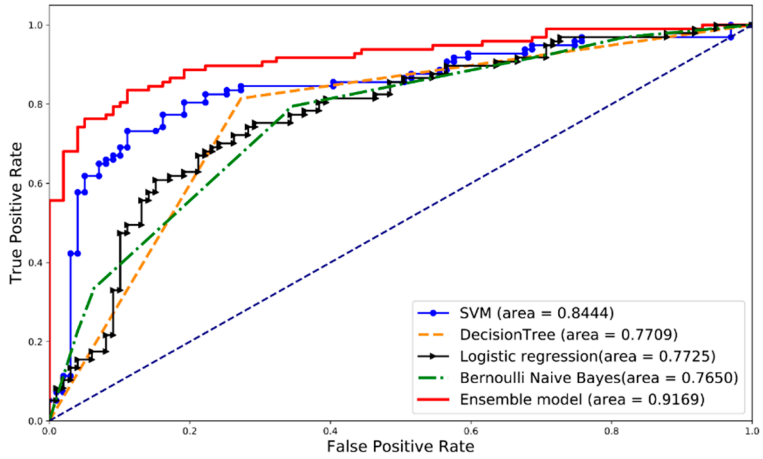


Figure 6. Receiver operating characteristic curve of the proposed model and other compared algorithms.

We performed a comparison study between the proposed ensemble model and several popular ensemble algorithms, including gradient boosting, RF, AdaBoost, and XGBoost. The average results and ROC curves of 10 rounds are listed in Table 5 and Figure 7, respectively.

Table 5. Comparison between the proposed method and other ensemble algorithms.

Model	Mean±Std.Dev.				
	Precision	Recall	F1	Error	AUC
GradientBoosting	0.8295 ± 0.0430	0.8028 ± 0.0502	0.8144 ± 0.0315	0.1765 ± 0.0000	0.8901 ± 0.0222
RandomForest	0.8471 ± 0.0407	0.7997 ± 0.0588	0.8210 ± 0.0336	0.1832 ± 0.0003	0.8826 ± 0.0257

AdaBoost	0.7724 ± 0.0433	0.7835 ± 0.0428	0.7763 ± 0.0223	0.2296 ± 0.0000	0.8374 ± 0.0236
XGBoost	0.8365 ± 0.0389	0.8026 ± 0.0652	0.8170 ± 0.0355	0.1837 ± 0.0000	0.8910 ± 0.0246
Proposed model	0.8600 ± 0.0375	0.8508 ± 0.0431	0.8543 ± 0.0257	0.1439 ± 0.0000	0.9207 ± 0.0177

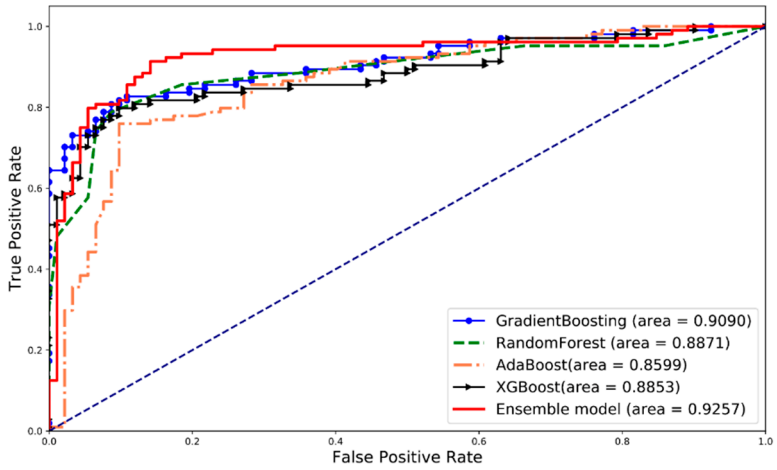


Figure 7. Receiver operating characteristic curves of the proposed model and other ensemble algorithms.

The results in Table 5 and Figure 7 show that the proposed model achieves the best performance on those five indicators. Next, an independent *t*-test was used to compare these models as shown in Table 6. The resulting differences between models were assumed as statistically significant when $p < 0.05$. These results show the statistically significant advantages of the proposed method compared to the other four models, particularly on *F1*, *Error*, and AUC. The proposed model is more efficient than AdaBoost on $Precision$ and $Recall$.

Table 6. Comparison of *p*-value between the proposed model and other ensemble algorithms.

Model	*p*-Value (This study vs. Other Algorithms)				
	Precision	Recall	F1	Error	AUC
GradientBoosting	0.1277	0.0568	0.0114	0.0000	0.0053
RandomForest	0.6565	0.0680	0.0441	0.0000	0.0022
AdaBoost	0.0004	0.0072	0.0000	0.0000	0.0000
XGBoost	0.2401	0.1004	0.0274	0.0000	0.0102

5.5. Feature Importance Analysis

To enhance the interpretability of the model, we carried out a feature importance analysis in this study. The contribution of each variable to the prediction performance is shown in Table 7. A feature importance analysis was applied to identify significant features based on a real data set. The Base-learners generate feature importance scores in different ways. The feature importance of the linear SVM is indeed the weight vector, which contains the coefficients, and these coefficients define an orthogonal vector to the hyperplane. The RF model measures the importance of the feature by calculating the corresponding out-of-bag (OOB) error. The AdaBoost generates feature scores by computing the normalized total reduction in the mean squared error, which is brought about by that feature with the sum of all feature importance levels equaling to one. The results of these methods are normalized, and the average value of the three normalized scores is the final level of feature importance. Figure 8 shows the ranking of the important features computed from different models, and the features are shown in different colors.

Table 7. The contribution of each variable to the prediction.

Features	SVM	RandomForest	AdaBoost	Averaged Score

c_GPA	0.0640	0.1331	0.1800	0.5221
competition_g	0.1639	0.0572	0.0200	0.0730
competition_n	0.7489	0.1538	0.1800	0.6804
competition_t	2.3491	0.1190	0.0400	0.5954
enrollment_c	0.5936	0.0764	0.0400	0.2204
gender	0.4920	0.0394	0.0200	0.0679
peer_b	1.3881	0.0867	0.0200	0.3337
pre_c_GPA	0.1086	0.1461	0.2400	0.6574
source_l	0.0240	0.0809	0.0800	0.2128
speciality	0.0163	0.1068	0.1800	0.4387

According to the average scores shown in Figure 8, *competition_n*, *pre_c_GPA*, *competition_t*, and *c_GPA* are the top four for feature importance. The results show the student's competition behavior and GPA play the most important roles in the model prediction performance, while the academic background is less important.

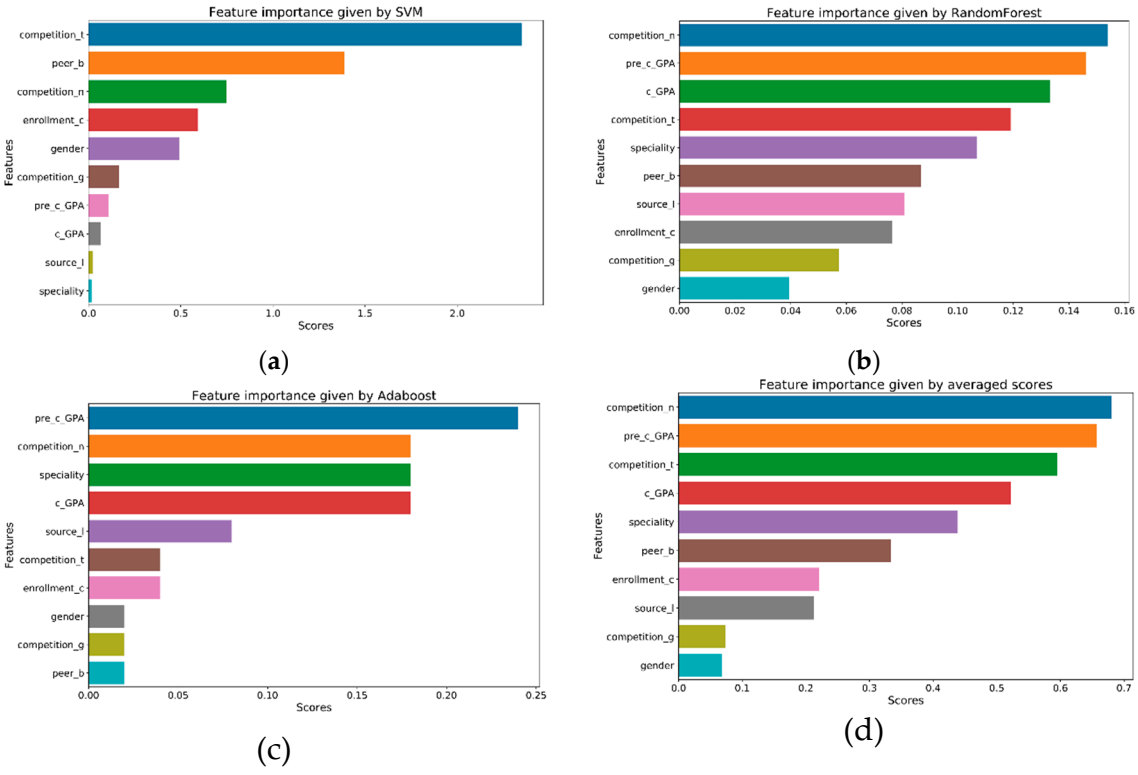


Figure 8. Levels of feature importance given by support vector machine (SVM) (**a**), random forest (**b**), AdaBoost (**c**), and the average values of scores from those three models (**d**).

6. Discussion and Conclusions

Predicting student academic performance has long been an important research topic. In this study, we propose an ensemble model using 2-layer stacking to predict student performance in academic competition. In this model, several algorithms (SVM, RF, and AdaBoost) with accurate prediction performance were implemented as the Base-learners, and a relatively simple algorithm (logistic regression) was adopted as the Meta-learner to reduce the risk of overfitting, and we used 10×10-fold cross-validation to avoid label leak.

The empirical research based on the data collected from Hankou University shows that the proposed ensemble model has better prediction performance compared to other models. Through the features importance analysis, we found that the competition behavior and GPA of candidates play the most important role in predicting competition results, and academic background is not as

the most important as expected. In the past, tutors and managers tended to pay more attention to students' academic background and GPA, but ignored their behavior in participating in competitions. To a certain extent, students' behavior in a competition can reflect students' cognitive understanding of competition and personality psychological characteristics. For example, the total number of participants in academic competitions can reflect a student's constancy of purpose in the competition. It is particularly important to pay more attention to the candidate's competition behavior for competition candidate selection in the future.

In 2019, we used this model to assist the student selection for the Blue Bridge Cup (National Software Professional Talent Design and Entrepreneurship Competition sponsored by the Talent Exchange Center of the Ministry of Industry and Information Technology) in Hankou University. The Blue Bridge Cup is a programming competition that is held every year. The participants' performance is ranked from high to low. The top 10% is the first prize, 20% is the second prize, and 30% is the third prize. About 30,000 college students from all over the country participated in the competition in 2019. It is difficult to win the prize, and in previous years the number of participants and awards of Hankou University were not positive. We imported the student data of three grades (computer major) into the model, and applied the results of prediction and feature analysis to the potential student selection Finally, 32 students participated in the competition, and 11 of them won the prize in 2019. Among them, one student was awarded the first prize, two students were awarded the second prize, and the others won the third prize in 2019. The number of winners and the proportion of winners increased significantly compared to the competition results in previous years.

Some research points should be paid more attention in the future study. First of all, the model should be applied to other universities in the future to further examine the stability and generalization of the model. Secondly, at present, it is a binary classification model. With the increase of data volume, we can research multiple classification models. More categories of competition results can be classified in the future to get more meaningful conclusions. Finally, the paper uses the traditional measures, such as $Precision$, $Recall$, $F1$, $Error$, and AUC to evaluate the model performance. These are commonly used measures for classification problems in machine learning. In future work, it is necessary to find new suitable measures for specific education scenarios.

Author Contributions: Conceptualization, Y.S.L; methodology, L.J.Y and Y.S.L; resources, L.J.Y; software, L.J.Y; supervision, Y.S.L; validation, L.J.Y; writing—original draft, L.J.Y and Y.S.L All authors have read and agreed to the published version of the manuscript.

Acknowledgments: The authors are grateful to the Educational Informatization Research Center of Hubei, Central China Normal University for providing financial support and good facilities. Additionally, they are thankful to Hankou University for providing the data.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. Support Vector Machine (SVM)

The support vector machine classifier is a binary classifier algorithm that looks for an optimal hyperplane as a decision function in a high-dimensional space [35–37]. The support vector machine (SVM) is a two-class classification model. Its basic model is defined as the most spaced linear classifier in the feature space, that is, the learning strategy of support vector machine is to maximize the interval, and eventually can be transformed into quadratic optimization of a convex function.

SVM is the most common machine learning algorithm that can use kernel techniques [38], and SVM has good generalization performance in small sample training sets, but if the amount of data is large, SVM training time will be longer. The performance of the support vector machine mainly depends on the selection of kernel function [39], so for a practical problem, the question of how to select the appropriate kernel function according to the actual data model to construct the SVM algorithm is critical. At present, many kernel function parameters depend on manual selection, with a certain degree of arbitrariness. In different problem areas, kernel functions should have different forms and parameters [40]. Common kernel functions in SVM are listed as the following equations:

$$Polynomial: k(X_1, X_2) = (X_1^T X_2)^n \tag{A1}$$

$$RBF: k(X_1, X_2) = \exp\left(-\frac{\left||X_1 - X_2|\right|^2}{2\sigma^2}\right) \tag{A2}$$

$$Laplacian: k(X_1, X_2) = \exp\left(-\frac{\left||X_1 - X_2|\right|}{2\sigma^2}\right) \tag{A3}$$

$$Sigmoid: k(X_1, X_2) = tanh[a(X_1^T X_2) - b], a, b > 0 \tag{A4}$$

Appendix A.2. Random Forest

RF [41] refers to an ensemble learning method of training, classifying, and predicting sample data by using multiple decision trees whose outputs are aggregated by majority voting. RFs [27] do not need to assume data distribution, it can handle thousands of input variables without variable deletion. At the same time, it gives estimates of what variables are important in the classification.

The common algorithm for building RF is described as follows:

Step 1. Randomly select K features among the total m features, where $K \ll m$, then randomly select J samples among the total n samples;

Step 2. With the K features over the J samples, to calculate the node d using the best split point;

Step 3. Split the node into daughter nodes using the best split.

Step 4. Repeat the above steps from 1 to 3 until the number of nodes l reached.

Step 5. Build forest by repeating the steps 1 to 4 for q times so that q trees will be created.

Appendix A.3. AdaBoost

AdaBoost is best used to boost the performance of decision trees on binary classification problems. AdaBoost is sensitive to noisy data and outliers. Otherwise, it is less susceptible to the overfitting problem than most learning algorithms. The workflow of the AdaBoost algorithm to solve the binary classification problem can be described as follows:

Step 1. Initialize the weight distribution of the training data;

$$D(1) = (\omega_{11}, \omega_{12}, \cdots, \omega_{1m}); \omega_{1i} = \frac{1}{m}; i = 1,2, \cdots, m \tag{A5}$$

Step 2. Get the classifier $G_k(x)$ by train on a data set with a weight distribution D_k;

Step 3. Calculate the classification error rate of $G_k(x)$

$$e_k = P(G_k(x_i) \neq y_i = \sum_{i=1}^{m} \omega_{ki} I(G_k(x_i) \neq y_i) \tag{A6}$$

Step 4. Calculate the coefficients of $G_k(x)$

$$a_k = \frac{1}{2} log \frac{1 - e_k}{e_k} \tag{A7}$$

Step 5. Update the weight distribution of data set

$$\omega_{k+1,i} = \frac{\omega_{ki}}{Z_k} \exp\left(-a_k y_i G_k(x_i)\right); i = 1,2, \cdots, m \tag{A8}$$

where Z_k is a normalization factor defined as below:

$$Z_k = \sum_{i=1}^{m} \omega_{ki} \exp\left(-a_k y_i G_k(x_i)\right); i = 1,2, \cdots, m \tag{A9}$$

Step 6. For $k = 1, 2, \ldots, K$, repeat Step 2 to 5 until K weak classifiers are trained;

Step 7. Output the final classifier.

$$f(x) = sign(\sum_{k=1}^{K} a_k G_k(x)). \quad \text{(A10)}$$

Appendix A.4. Logistic Regression

Schumacher [32] pointed out Logistic regression is a good choice for the prediction of success in a course or program. It is the basic model of prediction of a dichotomous dependent random variable. Logistic regression describes the relationship between a dichotomous dependent variable and a set of predictor variables. The predictor variables may be either numerical or categorical (dummy variables). This model is used for the prediction of the probability of the occurrence of an event by fitting data to a logistic curve.

The logistic regression model can be expressed as:

$$logit(y) = c_0 + c_1x_1 + c_2x_2 + \cdots + c_kx_k \quad \text{(A11)}$$

where $(x_1, x_2, \cdots, x_k)$ are independent variables, y is the dependent variable. $(c_1, c_2, \cdots, c_k)$ are coefficients which are adjusted y using the maximum likelihood technique and $logit(y) = ln\frac{y}{1-y}$.

With a given numerical cutoff (default value is set to 0.5 usually), cases with probabilities above this value are categorized as 1 (success), whereas cases lower than this value are classified as 0 (failure). However, the general accuracy of logistic regression is not high and is easy to be underfitted so that it cannot deal with nonlinear features very well [31].

References

1. Van Nul, W.T.D.; Roach, V.A. Head to head: The role of competition in undergraduate education. *Anat. Sci. Educ.* **2015**, *8*, 404–412.
2. Campbell, H.W.J.R.; Walberg, H.J. The theory of a general quantum system interacting with a linear dissipative system. *Annals of Physics.* 2000, pp. 547–607.
3. Campbell, J.R.; Walberg, H.J. Olympiad studies: Competitions provide alternatives to developing talents that serve national interests. *Roeper Rev*. **2010**, *33*, 8–17.
4. Goldstein, D.; Wagner, H. After school programs, competitions school olympics, and summer programs. *Int. Handb. Res. Dev. Gift. Talent* **1993**, *33*, 593–604.
5. Urhahne, D.; Ho, L.H.; Parchmann, I.; Nick, S. Attempting to predict success in the qualifying round of the international chemistry olympiad. *High Abil. Stud*. **2012**, *23*, 167–182.
6. Sandeep, M.J. Early alert of academically at-risk students: An open source analytics initiative. *J. Learn. Anal.* **2014**, *1*, 6–47.
7. Bouzayane, S.; Saad, I. Weekly predicting the at-risk mooc learners using dominance-based rough set approach. *Lect. Notes Comput. Sci.* **2017**, *10254*, 160–169.
8. Botelorenzo, M.L.; Gomezsanchez, E. Predicting the decrease of engagement indicators in a mooc. In Proceedings of the Seventh International Learning Analytics & Knowledge Conference on—LAK, Vancouver, BC, Canada, 13–17 March 2017; pp. 143–147.
9. Kennedy, G.; Coffrin, C.; De Barba, P.; Corrin, L. Predicting success: How learners' prior knowledge, skills and activities predict mooc performance. In Proceedings of the Fifth International Conference on Learning Analytics and Knowledge, Poughkeepsie, NY, USA, 16–20 March 2015; pp. 136–140.
10. Mann, C.M.; Canny, B.J.; Lindley, J.M.; Rajan, R. The influence of language family on academic performance in year 1 and 2 mbbs students. *Med. Educ*. **2010**, *44*, 786–794.
11. Johns, M.W.; Dudley, H.A.F.; Masterton, J.P. The sleep habits, personality and academic performance of medical students. *Med. Educ*. **1976**, *10*, 158–162.
12. Carter, S.P.; Greenberg, K.; Walker, M.S. The impact of computer usage on academic performance: Evidence from a randomized trial at the united states military academy. *Econ. Educ. Rev.* **2017**, *56*, 118–132.
13. Ok, M.W.; Kim, W. Use of ipads and ipods for academic performance and engagement of prek12 students with disabilities: A research synthesis. *Exceptionality* **2017**, *25*, 54–75.
14. Huang, S.; Fang, N. Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. *Comput. Educ*. **2013**, *61*, 133–145.

15. Al-Ghamdi, S.A.; Al-Bassiouni, A.A.M.; Mustafa, H.M.H.; Al-Hamadi, A. Simulation of improved academic achievement for a mathematical topic using neural networks modeling. *World Comput. Sci. Inf. Technol. J.* **2013**, *3*, 77–84.
16. Kotsiantis, S.B.; Pierrakeas, C.; Pintelas, P.E. Predicting students' performance in distance learning using machine learning techniques. *Appl. Artif. Intell.* **2004**, *18*, 411–426.
17. Romero, C.; Espejo, P.G.; Zafra, A.; Romero, J.R.; Ventura, S. Web usage mining for predicting final marks of students that use moodle courses. *Comput. Appl. Eng. Educ.* **2013**, *21*, 135–146.
18. Parikh, D.; Polikar, R. An ensemble-based incremental learning approach to data fusion. *Syst. Man Cybern.* **2007**, *37*, 437–450.
19. Beemer, J.; Spoon, K.M.; He, L.; Fan, J.; Levine, R.A. Ensemble learning for estimating individualized treatment effects in student success studies. *Artif. Intell. Educ.* **2018**, *28*, 315–335.
20. Ade, R.; Deshmukh, P.R. An incremental ensemble of classifiers as a technique for prediction of student's career choice. In Proceedings of the 2014 First International Conference on Networks & Soft Computing (ICNSC2014), Guntur, India 19–20 August 2014; pp. 384–387.
21. Kotsiantis, S.B.; Patriarcheas, K.; Xenos, M.N. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowl. Based Syst.* **2010**, *23*, 529–535.
22. Kearns, M.J.; Li, M.; Valiant, L.G. Learning boolean formulas. *J. ACM.***1994**, *41*, 1298–1328.
23. Schalk, P.D.; Wick, D.P.; Turner, P.R.; Ramsdell, M.W. Predictive assessment of student performance for early strategic guidance. In Proceedings of the 2011 Frontiers in Education Conference (FIE), Rapid City, SD, USA, 12–15 October 2011.
24. Hardman, J.; Paucar-Cáceres, A.; Fielding, A. Predicting students' progression in higher education by using the random forest algorithm. *Syst. Res. Behav. Sci*. **2013**, *30*, 194–203.
25. Shamsi, M.S.; Lakshmi, J. Student performance prediction using classification data mining techniques. *Arxiv: Learn.* **2016**. arXiv: 1606.05735.pdf.
26. Ishizue, R.; Sakamoto, K.; Washizaki, H.; Fukazawa, Y. Student placement and skill ranking predictors for programming classes using class attitude, psychological scales, and code metrics. *Res. Pract. Technol. Enhanc. Learn*. **2018**, *13*, 7.
27. Petkovic, D.; Sosnickperez, M.; Okada, K.; Todtenhoefer, R.; Huang, S.; Miglani, N.; Vigil, A. Using the random forest classifier to assess and predict student learning of software engineering teamwork. In Proceedings of the 2016 IEEE Frontiers in Education Conference (FIE), Eire, PA, USA, 12–15 October 2016; pp. 1–7.
28. Noori, R.; Karbassi, A.; Moghaddamnia, A.; Han, D.; Zokaeiashtiani, M.H.; Farokhnia, A.; Gousheh, M.G. Assessment of input variables determination on the svm model performance using pca, gamma test, and forward selection techniques for monthly stream flow prediction. *J. Hydrol.* **2011**, *401*, 177–189.
29. Han, M.; Tong, M.; Chen, M.; Liu, J.; Liu, C. Application of ensemble algorithm in students' performance prediction. In Proceedings of the 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan, 9–13 July 2017; pp. 735–740.
30. Poh, N.; Smythe, I. To what extend can we predict students' performance? a case study in colleges in south africa. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Orlando, FL, USA, 9–12 December 2014; pp. 416–421.
31. Allison, P.D. Logistic Regression Using the SAS System: *Theory Application. Jouranl of Chemical Information and Modeling.* **2019,** *53*, pp.1689-1699.
32. Schumacher, M.; Rosner, R.; Vach, W. Neural networks and logistic regression. *Comput. Stat. Data Anal.* **1996**, *21*, 661–682.
33. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai* **1995**, *14*, 1137–1145.
34. Fawcett, T. An introduction to roc analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874.
35. Boser, B.E.; Guyon, I.; Vapnik, V. A Training Algorithm for Optimal Margin Classifiers. *proceedings of annual acm workshop on computational learning theory,* **2008**, pp. 144–152. Available at: http://www.gautampendse.com/projects/bsvm/webpage/boser1992.pdf (accessed on 29 March 2020)
36. Vapnik, V. *Statistical Learning Theory*. Willy, 16, 1998. Available at: http://read.pudn.com/downloads161/ebook/733192/Statistical-Learning-Theory.pdf (accessed on 29 March 2020)

37. Cristianini, N.; Shawetaylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press*; 2000. Available at: https://books.google.com.hk/books?hl=en&lr=&id=_PXJn_cxv0AC&oi=fnd&pg=PR9&dq=37.%09Cristianini,+N.%3B+Shawetaylor,+J.+An+Introduction+to+Support+Vector+Machines+and+Other+Kernel-Based+Learning+Methods++.+Cambridge+University+Press&ots=xSUk6D-r09&sig=cO32--yeujiGuwGA8wHfqWbnAOU&redir_esc=y&hl=zh-CN&sourceid=cndr#v=onepage&q=37.%09Cristianini%2C%20N.%3B%20Shawetaylor%2C%20J.%20An%20Introduction%20to%20Support%20Vector%20Machines%20and%20Other%20Kernel-Based%20Learning%20Methods%20%20.%20Cambridge%20University%20Press&f=false (accessed on 29 March 2020)
38. Burges, C.J.C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167.
39. Aluko, R.O.; Daniel, E.I.; Oshodi, O.S.; Aigbavboa, C.; Abisuga, A.O. Towards reliable prediction of academic performance of architecture students using data mining techniques. *J. Eng. Des. Technol.* **2018**, *16*, 385–397.
40. Frohlich, H.; Chapelle, O.; Scholkopf, B. Feature selection for support vector machines by means of genetic algorithm. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, CA, USA, 3–5 November 2003; pp. 142–148.
41. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32.