SIN-CHN SCIENTIFIC PRESS PTE.LTD.

Molecular & Cellular Biomechanics 2025, 22(5), 692.
https://doi.org/10.62617/mcb692

Article

Educational data mining for student performance prediction in artificial intelligence environment

Linqiang Tang*, Sian Chen

Zhejiang Institute of Communications, Hangzhou 311112, China
*** Corresponding author:** Linqiang Tang, tlq87451@outlook.com

CITATION

Tang L, Chen S. Educational data mining for student performance prediction in artificial intelligence environment. Molecular & Cellular Biomechanics. 2025; 22(5): 692. https://doi.org/10.62617/mcb692

ARTICLE INFO

Received: 1 November 2024
Accepted: 12 November 2024
Available online: 24 March 2025

Abstract: In education, with the application of these information technologies, massive student data continue to produce, in order to realize the processing of data information, the traditional data mining technology is applied to the mass of education data processing process derived from a new technology, that is, education data mining technology. Among them, student performance prediction is an important application direction in education data mining, can help teachers to optimise their teaching decisions and help students to improve their learning plans. However, as of now, most of the models for student performance prediction suffer from weak generalization ability and poor feature correlation. Therefore, this paper proposes a student performance prediction method based on feature selection and Bagging integrated learning, which analyzes the model and a single prediction model, effectively solves the problem of low prediction accuracy of a single model, and improves the ability of the model to deal with the unseen examples to a certain extent, with a strong generalization ability.

Keywords: AI; student performance prediction; EDM; feature selection; bagging ensemble learning

1. Introduction

In the era of artificial intelligence, Internet information technology has been fully integrated into all corners of the natural society, which has a profound impact on people's production and life. For education and teaching, the massive student data that comes with the process of informatization is one of the most valuable resources in intelligent education, and the use of EDM (Educational Data Mining) to analyze a large amount of student data can be screened and construct a valuable knowledge network for teaching and learning, in order to better optimize the learning effect, assist in educational decision-making, and accelerate the process of the development of intelligent education. In the field of education, EDM can deeply analyze and explore the massive student and teacher data from the field of education through data statistics, vertical and horizontal cross-comparison, feature extraction, visual analysis and other methods, to obtain more intuitive and effective information about education knowledge, so as to better grasp the learning situation of students and the learning environment, and ultimately achieve the purpose of improving the education environment and enhancing the quality of teaching and learning.

Under the background of deepening education reform, EDM technology, which is formed by integrating education big data and data mining, has been studied by more and more scholars. Tang (2000) explored the data from the online education platform, identified the virtual knowledge structure in distance education by using WEB mining technology, and provided personalized learning plans for students who participated in

online courses by using online knowledge structure [1]. Romero.C. (2013) used student online data to predict final grades for 114 students in first-year computer science courses [2]. Hu (2014) using data-driven techniques to identify at-risk students, he found that temporal characteristics were key features in predicting student academic performance [3]. Pena-Ayala (2014) made an overall analysis of the research results related to EDM in the last three years, not only summarized the application mode of EDM methods, but also critically concluded that most EDM methods are based on three components, namely, rules, tasks, and algorithms [4]. In order to obtain a high-performance student attrition rate prediction model, Thammasiri (2014) cross-combined various sampling techniques with various classification regression algorithms, and verified the performance of the model on 9 evaluation indicators, so as to find the best-performing fusion prediction model [5]. Mubarak (2021) MOOC online student data was explored using convolutional neural networks to predict whether each student would drop out or complete the course [6].

In terms of student performance prediction research, Oyelade (2010) used K-means clustering algorithm to model the scores of 79 students in 9 previous courses, and predicted the scores of students according to their achievement levels [7]. Agaoglu (2016) used DTC5.0, SVM, ANN and discriminant analysis algorithms to analyze and model students' course evaluation data, and extracted the factors of students' behavior that affected teachers' teaching [8]. Ren (2016) based on the MOOC massive online learning platform, extracted a variety of characteristic data such as conversations, homework completion times and video learning duration generated by students during their learning, and built a student performance prediction model by using multiple regression algorithms [9]. This model can track students' learning participation in real time according to the click behavior recorded by the MOOC, and predict students' performance in the next placement exam in time. Gamulin (2016) used the idea of discrete Fourier transform to make a time series analysis of students' course log files, obtained the Times and time series of students' visits to an online course during their study, and established a prediction model for students' final grades based on this [10]. Tripathi (2019) used NB algorithm to model students' question-answer data set and predicted students' subjective and objective exam scores [11]. Compared with SVM algorithm, this prediction model showed better performance in execution time and accuracy. Tang (2023) built a multi-model fusion student performance prediction method based on the data of 103 undergraduates' Internet browsing logs, teaching activities and other data [12].

Scholars have conducted in-depth research on student performance prediction using artificial intelligence algorithms, and have achieved fruitful research results, showing that there is a strong relationship between various student attributes and their recorded activities in student management systems and their academic performance, and that most of the proposed prediction models have achieved significant results [13]. However, as a whole, the existing studies mainly focus on the task of student performance prediction for specific regional offline environments or specific online learning systems, which is highly targeted, with weak generalisation of the models and little attention to the hidden information of student attributes. Based on this, this paper introduces feature selection and Bagging integrated learning to conduct an in-depth study for this problem.

The main innovations : in the current research on student performance prediction, few scholars have addressed the impact of the correlation between student attribute features on the prediction model. Therefore, this paper proposes a student performance prediction method based on feature selection and Bagging integration learning, which improves the generalization ability of the algorithm through continuous feature selection and Bagging integration. The model uses three feature selection algorithms combined with Bagging model to train three processes sequentially, and the student performance prediction set generated by each process training is superimposed and updated in series form as model inputs; the Bagging method is again used to integrate and construct the model, and the ideal prediction algorithm is obtained.

2. Modeling framework based on feature selection and integrated learning

In the practical application of student performance prediction, although a single prediction model can accomplish the prediction purpose, it often has certain limitations in prediction ability, and it is difficult to obtain the in-depth knowledge information in the data space in the prediction task, which may cause unsatisfactory prediction effect of the constructed model or poor performance of the model and other problems. The FS-Bagging student performance prediction model proposed in this paper contains a four-layer structure, with the first three layers each consisting of a feature selection method and a Bagging integration architecture, and each layer corresponds to a prediction result after the training is completed, which serves as an input to the fourth layer structure, and then the Bagging integration learning is utilized to construct the model again, finally realizing the prediction of student performance. The model is shown in **Figure 1** below.

The FS-Bagging model mainly contains four core modules: Bagging integration learning algorithm, Chi-square Test, MRMR (Minimum Redundancy Maximum Relevance) and ReliefF. In the prediction model, a total of four Bagging integration processes are experienced, each time the integration process is the same, the specific steps are as follows:

(1) Bootstrap method was used for random sampling. Some returned training samples were extracted to form an independent sampling set of *t* group.

(2) Based on *t* sample sets, *t* base learner h_1, h_2, ..., h_t, the base learner function expression is as follows:

$$h_t = \Gamma(D, D_{bs}) \tag{1}$$

In the formula, D is the training set of the feature set, i.e. $D = \{(x_1, y_1), \ldots, (x_m, y_m)\}$.

(3) By combining *t* base learners with voting strategies, a strong learner is obtained, which is a prediction model *I*. The expression of the combination strategy function is:

$$H(x) = c_{argmax \sum_{i=1}^{T} w_i h_j^i(x)} \tag{2}$$

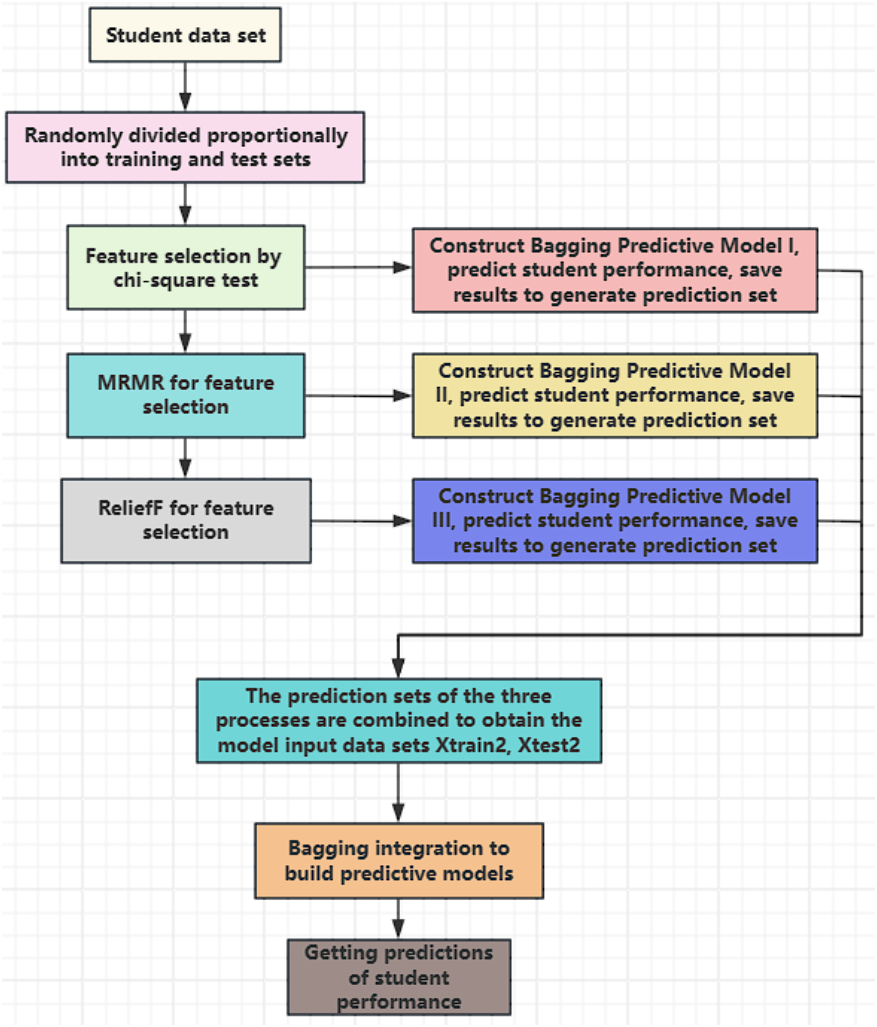

Figure 1. FS-Bagging student performance prediction model framework.

In the formula, w_i is the weight of h_i, usually $w_i \geq 0$, $\sum_{i=1}^{T} w_i = 1$.

(4) Predict student performance, output prediction label, save the result to generate prediction set.

3. Feature selection method

3.1. Chi-square test

The core idea of the chi-square test feature selection algorithm is to calculate the score value scores of each feature based on the *p*-value obtained from the chi-square test of different features and labeling classes, and then rank the score values of scores to select the more advanced features [16]. Using this algorithm to generate the feature set mainly includes the following steps:

(1) Determine the proportion of the total number of features selected from the original data set;

(2) Calculate the chi-square value χ^2 of each feature in the complete data set and the sample freedom *df*.

$$\chi^2 = \sum \frac{(A-E)^2}{E} \tag{3}$$

In the formula, *A* is the observed frequency and *E* is the expected frequency. Here, the feature in the sample corresponds to the observed value, and the label class corresponds to the theoretical value, that is, *A* is the sample feature, *E* is the label class, χ^2indicating the degree of correlation between feature x_i and class C;

(3) The scores obtained from each feature are calculated.

$$scores = -log(-p) \tag{4}$$

In the formula, p is the parameter to judge the degree of correlation between variables in the Chi-square test, and the value of p indirectly reflects the degree of correlation between sample features and label classes.

(4) Based on the scores ranking, the top decay percentage of features are selected to form a feature set [17]. In this step, scores is the variable form of feature p value. The larger the value, the greater the correlation degree between sample feature and label class. Rank the features according to their sizes to get the most relevant features of the target number.

3.2. Minimum redundancy-maximum correlation

Minimum redundancy-maximum relevance feature selection algorithm gives the best decay ratio of features. Its a classical feature selection method in which a subset of features that are most relevant to the target variable and least redundant with each other are selected from the original feature set. The algorithm measures the relevance of each feature to the target variable by calculating the mutual information value between them and further reduces the redundancy between features by calculating the redundancy between each feature and the selected set of features. When performing feature selection, the algorithm selects features with maximum correlation and minimum redundancy based on the trade-off between the mutual information value and redundancy, first selecting the feature with the highest correlation with the target variable as the most important feature, then selecting the feature with the minimum correlation with the selected set of features from among the remaining features, and repeating the process until the required number of features are selected. Generating the feature set using this algorithm mainly consists of the following steps:

(1) Calculate the mutual information *I* (*x*, *c*) between the features and the label class, find out the decay features that are most closely related to the label class *c*, and get the maximum correlation feature set.

$$max\ D\ (S,c) \qquad D = \frac{I}{|S|}\sum_{x_i \in S} I(x_i, c) \tag{5}$$

In the formula, *I* represents the mutual information of two variables, x_i is the sample feature, and *c* is the category label. Using this formula, a feature subset S containing m features can be obtained [18].

(2) Eliminate redundant features in feature subset *S* to obtain the minimum redundancy feature set.

$$\min R(S) \qquad R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, c_i) \tag{6}$$

(3) Find the feature set FS_2 about maximum relevance-minimum redundancy:

$$FS_2 = max[D - R]$$

$$FS = max[\frac{1}{|S|} \sum_{x_i \in S} I(x_i, c) - \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, c_i)] \tag{7}$$

In the formula, I is the mutual information of the two variables, x_i is the sample feature, and c is the category label.

3.3. Multi-class elimination

Multi-class elimination feature selection algorithm is essentially a feature weight algorithm [19]. Using this algorithm to generate feature set mainly includes the following sub-steps:

(1) A randomly selected sample in the feature set R;

(2) The k nearest neighbor samples are taken out within the sample group and the feature weights are calculated for them with the formula:

$$W_j^i = W_j^{i-1} - \frac{\Delta_j(x_r, x_q)}{m} \cdot d_{rq} \tag{8}$$

(3) The above steps are iterated m times, and the feature weights are updated in each iteration. Finally, the top features are selected according to the weight ranking to generate the feature set.

4. Experimental analysis

4.1. Build the data set

The student achievement data information used in this study was obtained from the UCI machine learning database, a dataset, which was collected from two schools in Hangzhou area. Real and valid questionnaire response data from 677 students were finally obtained [20]. These data were integrated into two datasets related to math and language to obtain the final dataset.

The dataset contains real questionnaire data of 677 students about 33 attributes, and the first 30 attributes are personal attributes of students, and the last three attributes are students' grades in the first, second and third terms, mainly involving the grades of the two main courses of Chinese and math [21]. If the records containing single subject grades are broken down to form the single subject data set, the final result is a data set of 395 students with only math grades and 649 students with only Portuguese grades.

4.2. Evaluation indicators

In order to verify the effectiveness of the model, this paper uses precision rate, recall rate, comprehensive evaluation index *f1-scores* and accuracy ACC to measure

the model performance from multiple dimensions. The calculation method of each evaluation index is as follows:

(1) Precision, which represents the percentage of accurate classification instances from all real classification instances, is calculated as follows:

$$Pr\,e\,cision = \frac{TP}{TP + FP} \tag{9}$$

(2) Recall, which measures the model's ability to recognize instances of positive classes, is calculated as:

$$Re\,c\,all = \frac{TP}{TP + FN} \tag{11}$$

(3) *f1-score*, which combines Precision and Recall of the model, is calculated as:

$$f_1 - score = \frac{2 \times Pr\,e\,cision \times Re\,c\,all}{Pr\,e\,cision + Re\,c\,all} \tag{12}$$

(4) Accuracy (ACC), an important indicator of how good a model is, is calculated as:

$$ACC = \sum_{n=1}^{n} \phi(i)/N \times 100\% \tag{13}$$

4.3. Result analysis

4.3.1. Effect of random segmentation sequence alignment between training set and test set on model accuracy

Before the experiment formally started, the initial data were first divided into training sets and test sets according to a certain proportion based on the randomized segmentation strategy [22]. Here, based on the math dataset and the language dataset, the prediction of whether a student can pass a certain course was made using the randomly generated training set and test set with different proportions, and the predicted ACC results of student performance for five random trainings were collected respectively, as shown in **Tables 1** and **2** below. In order to make it easier to see the optimal proportion of sequence division, the average of the accuracy obtained from the five random training sessions was compared as the model training results under different settings, as shown in **Figure 2** below.

Table 1. Prediction results of different proportion division under mathematical data set.

Training set: test set	First	Second	Third	Fourth	Fifth
9:1	92.5%	92.5%	90%	92.5%	90%
8:2	86.1%	89.9%	84.8%	87.3%	88.6%
7:3	89.9%	88.2%	91.6%	92.4%	90.8%
6:4	89.9%	89.2%	87.3%	89.9%	90.5%
5:5	89.9%	86.4%	88.9%	90.4%	89.4%

Table 2. Prediction results divided by different proportions under Chinese data set.

Training set: test set	First	Second	Third	Fourth	Fifth
9:1	93.9%	95.4%	93.9%	93.9%	92.3%
8:2	94.6%	90.8%	90.8%	92.3%	90.8%
7:3	90.8%	92.8%	92.8%	92.3%	93.3%
6:4	92.7%	92.7%	91.9%	91.2%	93.1%
5:5	92.9%	92.3%	90.5%	90.5%	91.1%

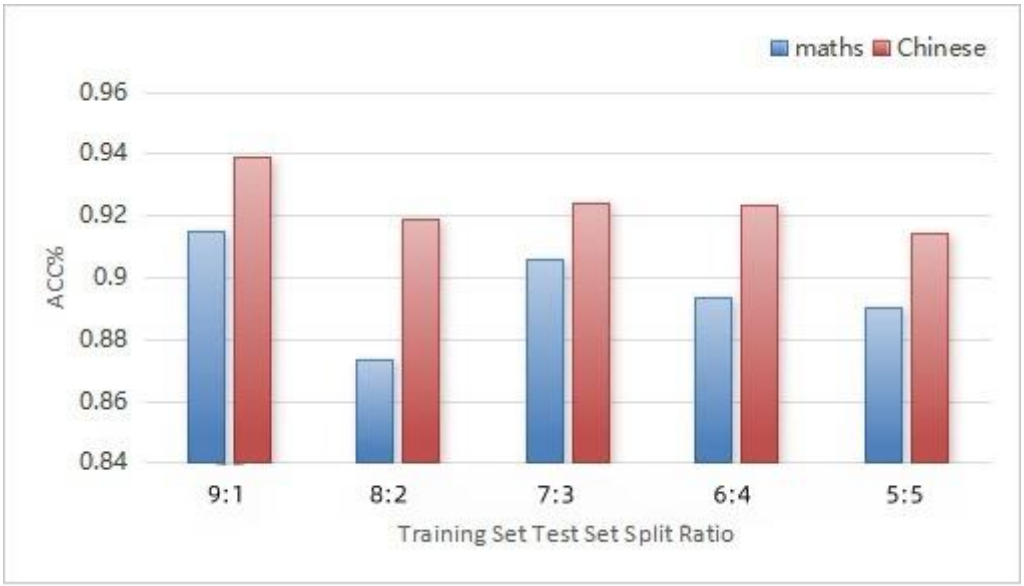

Figure 2. Dividing the predictions by different proportions.

When the size of the training and test sets is 9:1, the optimal prediction results are achieved both on the number dataset and on the language dataset [23]. Therefore, this experiment adopts a 9:1 random segmentation strategy to segment the original dataset to obtain the parameter optimization model for the prediction task.

4.3.2. Model performance test

The performance of the model of this paper was tested using mathematical datasets and the resulting evaluation results are displayed in **Table 3**.

Table 3. Performance evaluation of SF-Bagging algorithm under mathematical Settings.

Training Set (X)/Test Set (Y)	*P*	*R*	*f*1-*score*	ACC
$[X_1,Y_1]$	0.87	0.89	0.89	90%
$[X_2,Y_2]$	0.91	0.92	0.92	92.5%
$[X_3,Y_3]$	0.88	0.93	0.90	90%
$[X_4,Y_4]$	0.92	0.90	0.91	92.5%
$[X_5,Y_5]$	0.91	0.92	0.92	92.5%

The SF-Bagging model *f*1-*score* is stable at about 0.91, and the ACC is more than 90%. In order to more intuitively observe the changes of each index of the model, the

results of *f1-score* and ACC score were converted into histograms, as shown in **Figures 3** and **4** below.

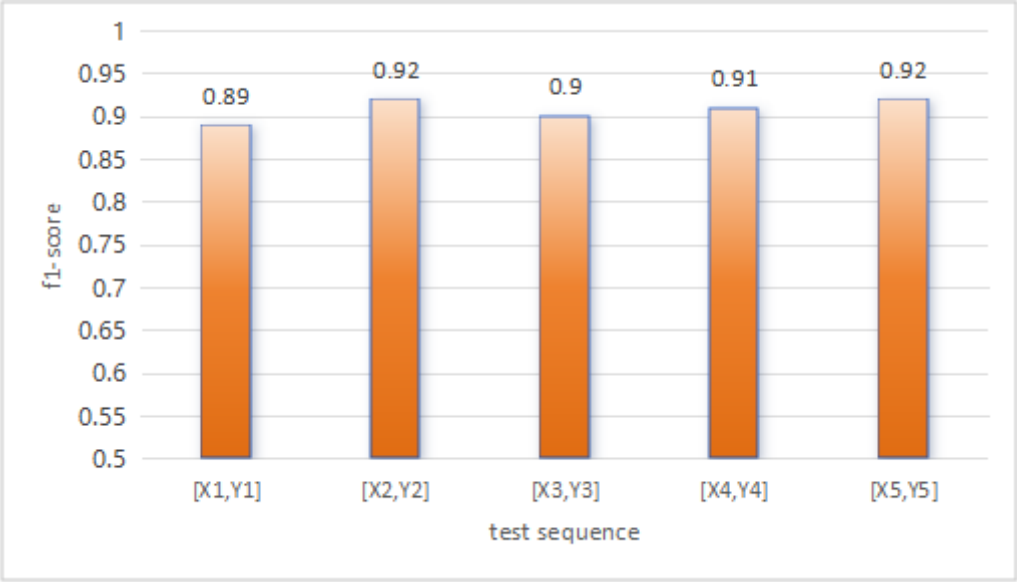

Figure 3. Evaluation results of *f1-score* under mathematical settings.

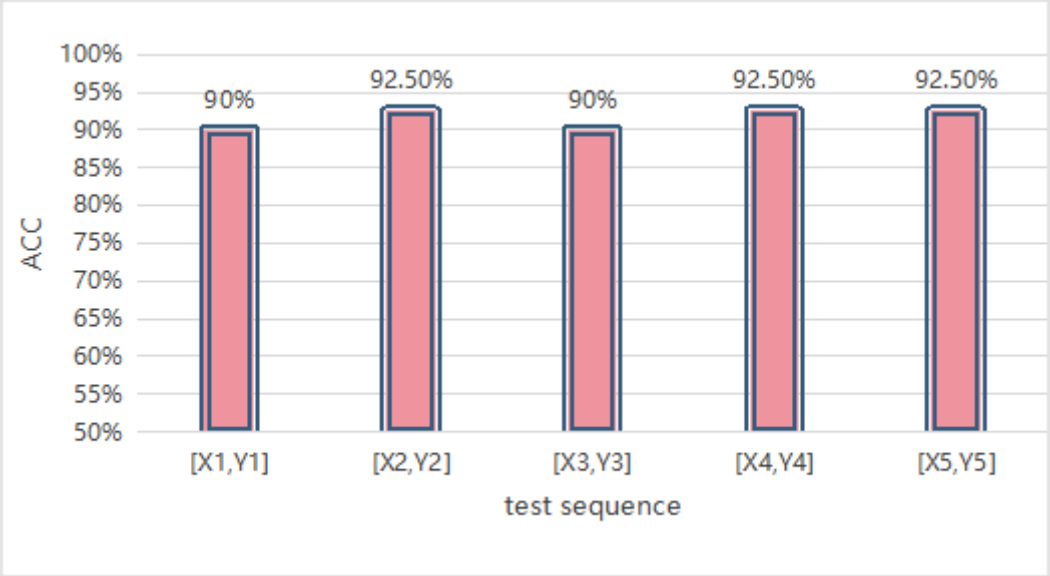

Figure 4. ACC evaluation results under mathematical settings.

Although random sequence partitioning is used to obtain the training and test sets, the performance of the model fluctuates around the mean value of each evaluation index in most cases, which is enough to show that the prediction model given in this paper has strong stability.

In order to further validate the model performance, here the model of this paper and DT, SVM, RF were compared and analyzed, using ACC and *f1-score* value as the basic evaluation indexes, five random experiments were conducted, the resulting four different algorithms based on the evaluation indexes of *f1-score* comparison results are shown in **Figure 5** below, and the algorithms based on the evaluation indexes of ACC derived from the accuracy of the algorithms are shown in **Figure 6** below.

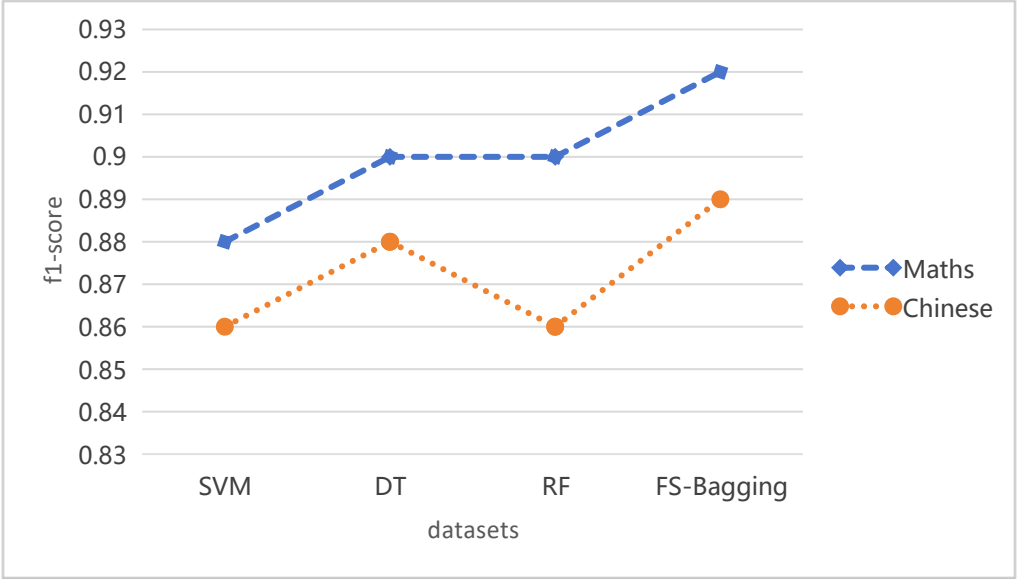

Figure 5. Values of different algorithms *f1-score*.

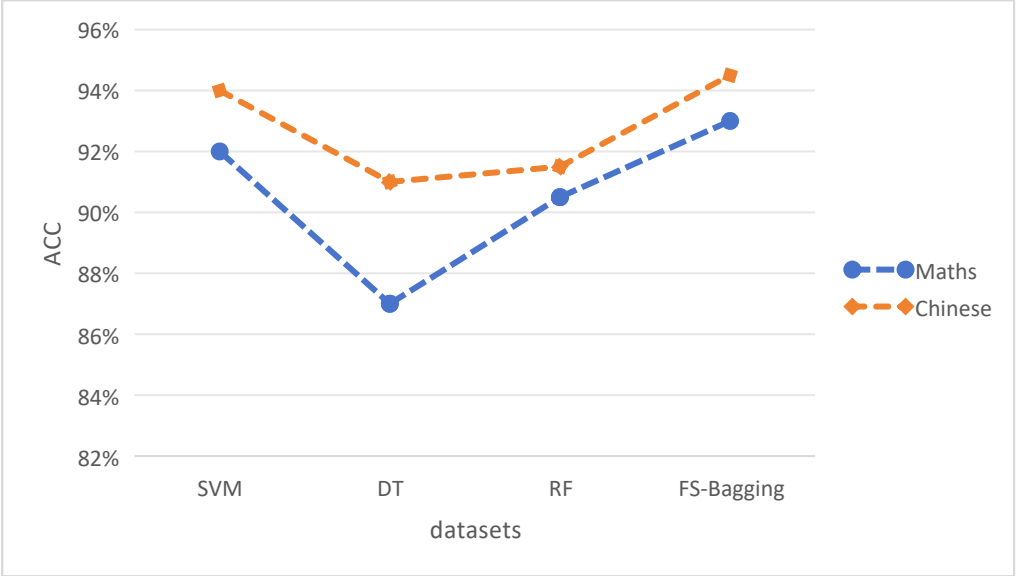

Figure 6. Accuracy of different algorithms.

It can be seen that for student performance prediction, all four algorithms show high accuracy and better comprehensive evaluation indexes. In terms of *f1-score* value, the comprehensive performance index of FS-Bagging algorithm is ahead of other algorithms by a small gap, and in terms of prediction accuracy, the accuracy of the algorithm proposed in this paper is significantly higher than that of the other single learners, which proves that compared with a single learner, the algorithm in this paper can better overcome the problems of lower performance of a single learner and easy to fall into the local extreme value point [25]. It can be seen that the FS-Bagging algorithm proposed in this paper has better performance and increases the correlation between features through feature selection and multi-layer combination calculation, which has obvious superiority in comparison.

5. Conclusion

In summary, this paper proposes a student performance prediction method based on feature selection and Bagging integration learning, which achieves model simplification and improves the algorithm's generalization ability through continuous feature selection and Bagging integration. The method samples the student dataset in a random sequence, obtains the training set and test set, and then performs feature selection followed by model training using the Bagging method and finally generates a prediction set for the prediction of student performance as a process, using the three feature selection algorithms to combine with the Bagging model to perform the three processes of training in turn, and the prediction set of student performance generated by each process of training is superimposed in the form of The prediction set of student performance generated by each process training is superimposed and updated in series form as model input; the Bagging method is again utilized to integrate and construct the model to obtain the ideal prediction algorithm. Finally, the experiments are compared with other existing models, and the analysis results show that this method can effectively improve the accuracy of student performance prediction. The reason for this is that compared with a single learner, the model in this paper can better overcome the problems of low performance of a single learner and easy to fall into local extreme points, so it has a more excellent performance. This study provides the possibility for students to have more timely and efficient learning interventions.

As a whole, this study for student performance prediction has achieved the established should research results, but there are still some shortcomings. For example, the dataset established in this paper has a small sample, and in the context of the development of more and more educational data, the results of the study are not enough to meet the demand for educational data mining. In the future research process, the author will actively collect data about more dimensions of students, use a larger dataset to verify the generalization ability of the model, and adopt more advanced mining techniques to conduct research on student performance prediction.

Conflict of interest: The authors declare no conflict of interest.

References

1. Changjie Tang, Rynson W.H. Lau, Qing Li, Huabei Yin, Tong Li, Danny Kilis. Personalized courseware construction based on Web data mining[C]. First International Confe- rence on Web Information Systems Engineering, Hong Kong, China, 2000, 2, 204-211.
2. Ya-Han Hu, Chia-Lun Lo, Sheng-Pao Shih. Developing early warning systems to predict students' online learning performance[J]. Computers in Human Behavior, 2014, 36:469-478.
3. Romero C, Lopez M I, Luna J M, et al. Predicting students' final performance from
4. participation in on-line discussion forums[J]. Computers & Education, 2013, 68:458-472.
5. Pena-Ayala A. Educational data mining: A survey and a data mining-based analysis ofrecent works[J]. Expert systems with applications, 2014, 41(4): 1432-1462.
6. Thammasiri D, Delen D, Meesad P, et al. A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition[J]. Expert Systems with Applications, 2014, 41(2): 321-330.
7. Oyelade O J, Oladipupo O O, Obagbuwa I C. Application of k-Means Clustering algorithm for prediction of Students Academic Performance[J]. International Journal of Computer Science & Information Security, 2010, 7(1): S39.

8. Mubarak Ahmed A., Cao Han, Hezam Ibrahim M.. Deep analytic model for student dropout prediction in massive open online courses[J]. Computers and Electrical Engineering, 2021,93.
9. Ren Zhiyun, Rangwala Huzefa, Johri Aditya. Predicting Performance on MOOC
10. Assessments Using Multi-Regression Models[P]. International Educational Data Mining Society. 2016.
11. Gamulin J, Gamulin O, Kermek D. Using Fourier coefficients in time series analysis for student performance prediction in blended learning environments[J]. Expert Systems, 2016,33.
12. Agaoglu M. Predicting instructor performance using data mining techniques in higher education[J]. IEEE Access, 2016(4): 2379-2387.
13. Tripathi A, Yadav S, Rajan R. Naive Bayes Classification Model for the Student Performance Prediction[C]. 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). 2019.
14. Tang, Xi. Analysis and Research on Students' Image Construction and Academic Situation Prediction Based on Educational Data Mining [J]. Modern Information Technology,2023,7(04): 193-198.
15. Albreiki B, Zaki N, Alashwal H. A systematic literature review of student'performance prediction using machine learning techniques[J]. Education Sciences, 2021, 11(9): 552.
16. Xu Z, Yuan H, Liu Q. Student performance prediction based on blended learning[J]. IEEE Transactions on Education, 2020, 64(1): 66-73.
17. Hashim A S, Awadh W A, Hamoud A K. Student performance prediction model based on supervised machine learning algorithms[C]//IOP conference series: materials science and engineering. IOP Publishing, 2020, 928(3): 032019.
18. Sekeroglu B, Abiyev R, Ilhan A, et al. Systematic literature review on machine learning and student performance prediction: Critical gaps and possible remedies[J]. Applied Sciences, 2021, 11(22): 10907.
19. Chitti M, Chitti P, Jayabalan M. Need for interpretable student performance prediction[C]//2020 13th International Conference on Developments in eSystems Engineering (DeSE). IEEE, 2020: 269-272.
20. Alamri R, Alharbi B. Explainable student performance prediction models: a systematic review[J]. IEEE Access, 2021, 9: 33132-33143.
21. Bilal M, Omar M, Anwar W, et al. The role of demographic and academic features in a student performance prediction[J]. Scientific Reports, 2022, 12(1): 12508.
22. Kishor K, Sharma R, Chhabra M. Student performance prediction using technology of machine learning[C]//International Conference on Micro-Electronics and Telecommunication Engineering. Singapore: Springer Nature Singapore, 2021: 541-551.
23. Pallathadka H, Wenda A, Ramirez-Asís E, et al. Classification and prediction of student performance data using various machine learning algorithms[J]. Materials today: proceedings, 2023, 80: 3782-3785.
24. Sravani B, Bala M M. Prediction of student performance using linear regression[C]//2020 International Conference for Emerging Technology (INCET). IEEE, 2020: 1-5.
25. Xu Z, Yuan H, Liu Q. Student performance prediction based on blended learning[J]. IEEE Transactions on Education, 2020, 64(1): 66-73.
26. Yang F, Li F W B. Study on student performance estimation, student progress analysis, and student potential prediction based on data mining[J]. Computers & Education, 2018, 123: 97-108.
27. Asselman A, Khaldi M, Aammou S. Enhancing the prediction of student performance based on the machine learning XGBoost algorithm[J]. Interactive Learning Environments, 2023, 31(6): 3360-3379.