

## Article

# A Method for Prediction and Analysis of Student Performance That Combines Multi-Dimensional Features of Time and Space

Zheng Luo <sup>1,†</sup>, Jiahao Mai <sup>1,†</sup>, Caihong Feng <sup>1</sup>, Deyao Kong <sup>1</sup>, Jingyu Liu <sup>1,\*</sup>, Yunhong Ding <sup>1,\*</sup>, Bo Qi <sup>2</sup> and Zhanbo Zhu <sup>3</sup>

<sup>1</sup> The School of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China; luozheng@stu.hrbnu.edu.cn (Z.L.); majiahao@stu.hrbnu.edu.cn (J.M.); fengcaihong@stu.hrbnu.edu.cn (C.F.); kongdeyao@stu.hrbnu.edu.cn (D.K.)

<sup>2</sup> School of Astronautics, Harbin Institute of Technology, Harbin 150001, China; 22b918100@stu.hit.edu.cn

<sup>3</sup> No. 703 Research Institute, China State Shipbuilding Corporation Limited, Harbin 150025, China; zzbb630@gmail.com

\* Correspondence: yushengyi@hrbnu.edu.cn (J.L.); dingyunhong@hrbnu.edu.cn (Y.D.)

† These authors contributed equally to this work.

**Abstract:** The prediction and analysis of students' academic performance are essential tools for educators and learners to improve teaching and learning methods. Effective predictive methods assist learners in targeted studying based on forecast results, while effective analytical methods help educators design appropriate educational content. However, in actual educational environments, factors influencing student performance are multidimensional across both temporal and spatial dimensions. Therefore, a student performance prediction and analysis method incorporating multidimensional spatiotemporal features has been proposed in this study. Due to the complexity and nonlinearity of learning behaviors in the educational process, predicting students' academic performance effectively is challenging. Nevertheless, machine learning algorithms possess significant advantages in handling data complexity and nonlinearity. Initially, a multidimensional spatiotemporal feature dataset was constructed by combining three categories of features: students' basic information, performance at various stages of the semester, and educational indicators from their places of origin (considering both temporal aspects, i.e., performance at various stages of the semester, and spatial aspects, i.e., educational indicators from their places of origin). Subsequently, six machine learning models were trained using this dataset to predict student performance, and experimental results confirmed their accuracy. Furthermore, SHAP analysis was utilized to extract factors significantly impacting the experimental outcomes. Subsequently, this study conducted data ablation experiments, the results of which proved the rationality of the feature selection in this study. Finally, this study proposed a feasible solution for guiding teaching strategies by integrating spatiotemporal multi-dimensional features in the analysis of student performance prediction in actual teaching processes.



**Citation:** Luo, Z.; Mai, J.; Feng, C.; Kong, D.; Liu, J.; Ding, Y.; Qi, B.; Zhu, Z. A Method for Prediction and Analysis of Student Performance That Combines Multi-Dimensional Features of Time and Space. *Mathematics* **2024**, *12*, 3597. <https://doi.org/10.3390/math12223597>

Academic Editor: Iraklis Varlamis

Received: 14 October 2024

Revised: 13 November 2024

Accepted: 15 November 2024

Published: 17 November 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Accurate prediction and analysis of students' academic performance are key steps in enhancing the quality of education. Advances in big data technology and the spread of educational informatization have led to the recording of a growing volume of multi-dimensional spatiotemporal data. This provides educators with abundant informational resources but also introduces new challenges [1]. Traditional prediction methods tend not to fully capture the intricate factors impacting students' academic performance, resulting in outcomes that may not genuinely benefit students and teachers [2]. To effectively identify

underperforming students and provide timely assistance, enhancing the quality of education and learning outcomes requires reliance on quantitative analysis and methods for assessing academic performance. To uncover the complex relationships between academic performance and its influencing factors, data mining techniques have been applied in education, leading to the emergence and development of Educational Data Mining (EDM) [3]. As a specialized branch of data mining, EDM focuses on extracting valuable information from educational data to gain insights into teaching and learning patterns and support educational decision-making. Educational institutions globally aim to address issues of poor student performance and premature dropout [4]. Prediction of Student Academic Performance (SAP) [5] has long been a focus for educators, as it is essential for proactive measures to improve learning and reduce the risk of student dropout and failure. SAP prediction involves analyzing various influencing factors. Current research mainly depends on educational information stored in management systems, in the form of student records, to predict academic performance, including students' basic information and academic background [6]. Mengash et al. assembled a dataset of 2039 students enrolled in a public university's Computer Science and Information faculty in Saudi Arabia from 2016 to 2019, employing data mining techniques to aid the university's admission decisions and predict applicants' academic performance [7]. Baruah et al. proposed a method for predicting student performance using the MapReduce framework [8]. Feng et al. determined the number of clusters in the K-means algorithm and applied discriminant analysis to test the clustering for analyzing and predicting students' academic performance [9]. Liu, along with Wang et al., trained a feedforward spiking neural network with data from educational management systems and online learning platforms to predict students' academic performance [10]. Yue et al. suggested the use of a multi-objective grey wolf optimizer (GWO) with cost-sensitive feature selection for predicting students' academic performance in university-level English [11].

With rapid advancements in artificial intelligence and computer science, numerous advanced AI technologies have found broad application within engineering systems. Concurrently, machine learning has attracted considerable attention in Educational Data Mining (EDM) and has been utilized extensively for predicting students' academic performance. A system employing multiple splits based on the Gini index and  $p$ -values was used by Noor et al. to build models combining six potential base machine learning algorithms arbitrarily [12]. Bansal et al. created an automated evaluation system for student performance using deep learning and machine learning techniques during the COVID-19 pandemic [13]. Asselman et al. suggested a PFA method utilizing different models like Random Forest, AdaBoost, and XGBoost to predict students' academic performance [14]. The effectiveness of machine learning algorithms, including Naive Bayes, ID3, C4.5, and SVM, in forecasting students' academic performance based on prior course performance was investigated by Pallathadka et al. [15]. Zhang et al. implemented a federated learning strategy, tailoring analytics to utilize distributed data while preserving privacy and decentralization [16]. To improve the accuracy of predicting students' academic performance, machine learning algorithms have become an indispensable part of the EDM field and play an increasingly important role in educational evaluation. However, most current predictions of students' academic performance lack consideration of multidimensional spatiotemporal factors. Firstly, in the learning of first-year courses, due to disparities in educational resources across different regions, students from different places of origin have varying levels of foundational grasp of the courses, thus experiencing different levels of difficulty in learning [17]. Secondly, most current research on predicting students' academic performance does not track students' performance over time, neglecting changes during the learning process. Unlike previous studies, in this research, a comprehensive multidimensional spatiotemporal feature framework was constructed to predict student performance. Innovatively, the combination of student basic information, learning performance at different stages within a semester, and educational indicators from the students' origin regions was incorporated. This integration takes into account the dynamism of the student learning

process as well as the potential impact of regional educational disparities on academic performance. Firstly, differences in educational resources across regions result in varying levels of foundational knowledge mastery among students upon enrollment, which subsequently affects their performance at various stages of university courses. By incorporating these spatiotemporal factors, researchers can delve deeper into the complex mechanisms influencing student performance, providing a more solid foundation for accurate prediction and effective intervention.

Educational data, such as educational indicators from the students' regions of origin, were utilized in this study, which have often been overlooked in previous predictions of student performance. However, in practical educational activities, this is indeed an important factor that cannot be ignored. Taking the research subjects (first-year C programming course) used in this study as an example, China, as a vast country, exhibits significant regional imbalances in educational development [18]. For instance, students from economically developed regions such as Beijing, Shanghai, and Guangdong, compared with those from regions with relatively scarce educational resources, such as western and northeastern areas, show considerable differences in computer education. This disparity is particularly pronounced during the early stages of undergraduate education, leading to students from regions with abundant computer educational resources being more likely to adapt to and master the course content, whereas students from regions with scarce computer educational resources may face greater learning challenges. In addition, data on student performance at each stage of the semester were collected in this study. While using students' classroom performance and scores throughout the learning process to predict academic performance is not uncommon in previous studies, the dynamic nature of student performance over a semester is often neglected [19,20]. By collecting data on student performance at all stages of the semester (including homework scores, exam scores, and laboratory scores), this study aimed to avoid the loss of key information for predicting academic performance to a certain extent. Moreover, the data on student performance at each stage of the semester represent a temporal feature, indicating that final outcomes could be predicted even as data are being collected (for example, only the first and second stage performance data of students might be available early in the course, but this is sufficient to model the system and make a preliminary prediction of the student's final grade). This allows teachers to predict student grades at an earlier stage of the course and take targeted measures accordingly.

This paper utilizes diverse machine learning techniques to grasp intricate connections among academic outcomes and multiple space-time elements. Various machine learning approaches' effectiveness is examined, identifying key features that influence forecast accuracy. The structure of the subsequent sections is as follows. Firstly, dataset creation and preparation techniques are outlined. Secondly, the conceptual framework, foundational theory, overarching architecture, and practical steps for the machine learning-driven academic outcome prediction model are described. Following this, the predictive model leveraging machine learning is deployed on an education dataset rich with multidimensional space-time characteristics, exploring the significance of individual attributes on the findings. Lastly, the research draws its conclusions.

The achievements of this research include:

1. Development of an educational dataset incorporating multidimensional space-time attributes aimed at forecasting student performance.
2. Successful application of a predictive model based on a dataset enriched with multidimensional space-time attributes for forecasting student performance.
3. Examination of attribute significance in relation to prediction outcomes, pinpointing critical elements impacting student performance.
4. Based on the prediction results and feature importance analysis, this study proposes a dynamic optimization strategy for enhancing teaching and learning behaviors in actual educational processes. It provides a guiding and feasible solution for the participants in educational activities to use the educational information generated during the prediction and analysis process.

## 2. Dataset and Data Preprocessing

### 2.1. Dataset Construction

#### 2.1.1. Data Source

The C programming language course is a classic course for freshmen in computer science majors. This study collected data from 300 students across three cohorts (2021, 2022, and 2023) from the software engineering department of a teacher training university located in Northeast China. The data includes basic demographic information, scores from various stages of the first-semester C language course (including laboratory work and practical exams), and educational indicators from the students' places of origin. By integrating these three types of features, this study constructed a dataset that combines students' performance at different stages of the semester with educational indicators from their places of origin, as shown in Table 1. The average age of participants ranges from 18 to 21 years. To protect the privacy of the participants, all collected data have been de-identified to ensure that no information capable of identifying individuals is disclosed.

**Table 1.** Overall dataset.

Feature Category	Feature Name
Basic student data	Age Gender Class Nation Origin of student College entrance examination score Student category (urban/rural)
Performance data of students at each stage	Experimental score in stages 1 to 7 of the semester Term stage 1 to 7 test scores
Education indicators of student origin	Number of teachers with PhD degrees in the source area Number of teachers with master's degrees in the student regions Number of teachers with undergraduate degrees in the source region Number of senior teachers in the student source regions Number of deputy senior teachers in the origin Number of intermediate teachers in the origin Number of digital terminals in the student origin Number of multimedia classrooms in student places Student source of education investment assets Number of educational instruments and equipment in the student source area Total number of books collected in the student areas
label	Student course scores

#### 2.1.2. Basic Information of Students

Students' basic information includes gender, age, class, ethnicity, place of origin, student category, and college entrance examination scores. These features precede the students' engagement with the course and reflect the basic situation of each individual student. Naturally, there is a significant correlation between these characteristics and each student's ability to learn the course, which, in turn, influences the students' final scores in the course.

#### 2.1.3. Students' Performance at All Stages of the Semester

The performance of students at various stages throughout the semester can reveal learning trends and allow observations of progress at different points. Students who show continuous improvement typically perform better in their final examinations. By analyzing students' performance at different stages, it is possible to identify the phases that most significantly impact final grades. This insight can assist educators in focusing their efforts on those critical phases to enhance overall course performance and reduce failure rates.

In this study, the data reflecting students' performance at various stages of the semester were derived from scores obtained in assignments and laboratory work during the first-semester C programming language course at a teacher training university located in Northeast China. The semester was segmented according to seven chapters of the course material. Data were collected on students' assignment and lab scores for these seven phases, thereby constituting the features representing students' performance at different stages of the semester.

#### 2.1.4. Education Indicators of Student Origin

Due to the different sources of students, it is obvious that students are not exposed to college courses in middle school, especially in the first year, this gap is often significant. Therefore, at the beginning of the course, the educational indicators in the past decade are important. This study selected a number of provinces based on several indicators, including the number of teachers with PhDs, master's degrees, undergraduate degrees, senior teachers, deputy senior teachers, and intermediate teachers, the number of digital terminal stations, digital terminals, multimedia classrooms, education investment assets, education equipment, total number of books. These indicators were used to construct the student education index. The sources of the indicator data have been stated in the Data Availability Statement section.

#### 2.2. Data Processing

Firstly, individual student data inherently contain a degree of privacy. Secondly, since instructors do not deliberately retain students' learning data during their teaching, the educational data collected by researchers are often incomplete and limited. Additionally, due to imbalances in educational indicators from different places of origin and the tendency of schools' admission policies to favor local students, the data collected are often imbalanced. To address these issues, the experiment adopted the following measures.

This study categorized students' grades (G) into five levels ( $G < 60$  as 0,  $60 \leq G < 70$  as 1,  $70 \leq G < 80$  as 2,  $80 \leq G < 90$  as 3,  $90 \leq G \leq 100$  as 4). In actual teaching scenarios, the number of students in each grade level is often imbalanced, which can affect the predictive performance of the models.

To mitigate the effects of these imbalances, appropriate strategies, such as data augmentation, resampling techniques (such as oversampling the minority classes or undersampling the majority classes), or employing algorithms that are robust to imbalanced datasets, might be applied. These strategies help ensure that the model remains effective and reliable despite the uneven distribution of data across different grade levels.

- In addressing the issue of limited and incomplete data, this study employed a novel data imputation method to fill in missing values. Additionally, a new data augmentation technique was adopted to increase the volume of data, facilitating the training of predictive models.
- In the data processing stage, the category imbalance of students should be taken into account and processed.

#### 2.2.1. Normalization

To avoid the effect of dimension on model precision, the data were normalized. In this study, the min–max scaling algorithm was used for normalization [21].

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where  $\hat{x}$  represents the normalized attribute value.  $\min(x)$  denotes the minimum value in the attribute values, and  $\max(x)$  denotes the maximum value in the attribute values.

### 2.2.2. Missing Value Completion

KNN (K-Nearest Neighbors) interpolation is a method based on neighboring data points. This method involves finding the K nearest known data points to the target point and estimating the value of the target point based on the values of these neighbors [22]. The basic steps of KNN interpolation are as follows:

1. Select K value: determine the size of K, usually through cross-validation to select the best K.
2. Calculate Distance: calculate the Euclidean distance between the target point and all known points. The Euclidean distance formula is:

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^M (x_{i,m} - x_{j,m})^2} \quad (2)$$

where  $x_i$  and  $x_j$  are data points, and M is the dimension of the data.

3. Find K Nearest Neighbors: select the K known data points that are closest to the target point.
4. Weighted averaging: weight the values of K neighbors, with weights inversely proportional to distance. The weighted average interpolation for K nearest neighbors can be expressed as

$$\hat{y} = \frac{\sum_{k=1}^K w_k y_k}{\sum_{k=1}^K w_k} \quad (3)$$

where  $y_k$  is the value of the  $k - th$  neighbor and  $w_k$  is the weight of the distance from the point to be interpolated, which is usually defined as the inverse ratio of the distance:

$$w_k = \frac{1}{d_k} \quad (4)$$

### 2.2.3. Deal with Unbalanced Data

Synthetic Minority Oversampling Technique (SMOTE) is an interpolation method used to deal with unbalanced datasets, especially in classification tasks, where minority samples are oversampled [23]. SMOTE improves the performance of the classifier by synthesizing new minority samples to enhance the diversity of the dataset. The specific steps are as follows:

1. Select Minority Sample: select a random sample from the minority group.
2. Calculate neighbors: use a distance metric to find k nearest neighbors for that sample.
3. Generate a new sample: a neighbor is randomly selected from these k neighbors and a new sample is synthesized according to the following formula:

$$\text{new\_sample} = x_i + \lambda(x_j - x_i) \quad (5)$$

## 3. Research Methods

### 3.1. Related Technologies

#### 3.1.1. Predictive Models

In this study, six machine learning models were used to predict students' final grades: XGboost, LightGBM, random forest, AdaBoost, decision tree, and SVM.

##### 1. XGboost

XGBoost is an optimized boosting algorithm, which gradually improves the accuracy of the model by continuously splitting features to generate new trees to fit the residuals of the previous tree [24]. As a gradient boosting decision tree algorithm, the XGBoost algorithm model can be regarded as a decision tree additive strategy model, that is

$$\hat{y}_p = \sum_{t=1}^M f_t(x_p), f_t \in D \quad (6)$$

where  $x_p$  is the  $p - th$  input data,  $y_p$  is the corresponding prediction value,  $M$  is the number of decision trees,  $D$  is the set of decision trees, and  $f_t(x_p)$  is the corresponding  $t - th$  decision tree in space  $D$ . When the model performs iterative learning, it first initializes a predicted value, and adds a new function  $f$  for each iteration, and the iterative learning process can be expressed as

$$\begin{aligned}\hat{y}_p^{(0)} &= 0 \\ \hat{y}_p^{(1)} &= \hat{y}_p^{(0)} + f_1(x_p) \\ &\dots \\ \hat{y}_p^{(t)} &= \hat{y}_p^{(t-1)} + f_t(x_p)\end{aligned}\tag{7}$$

The objective function  $B_{Obj}$  of the XGBoost algorithm is not only used to measure the model fitting error, but also includes regularization terms to limit the complexity of each tree, which is expressed as

$$B_{Obj} = \sum_{p=1}^n l(y, \hat{y}) + \sum_{t=1}^M \Omega(f_t)\tag{8}$$

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda ||w||^2\tag{9}$$

In Equation (7),  $\sum_{p=1}^n l(y, \hat{y})$  is used to measure the error between the model's predicted values and the true values, while  $\sum_{t=1}^M \Omega(f_t)$  serves as a regularization term to penalize the complexity of the tree, preventing overfitting. In Equation (8),  $T$  represents the number of leaf nodes,  $\gamma$  is used to constrain the number of nodes, favoring simpler models;  $w$  represents the score of the leaf node, applying an L2 norm constraint to limit the estimated score of each leaf node. The regularization part in XGBoost is also one of the advantages of the algorithm. After the  $t - th$  iteration, the objective function can be written as

$$L^{(t)} = \sum_{p=1}^n l(y_p, \hat{y}_p^{(t-1)} + f_t(x_p)) + \Omega(f_t)\tag{10}$$

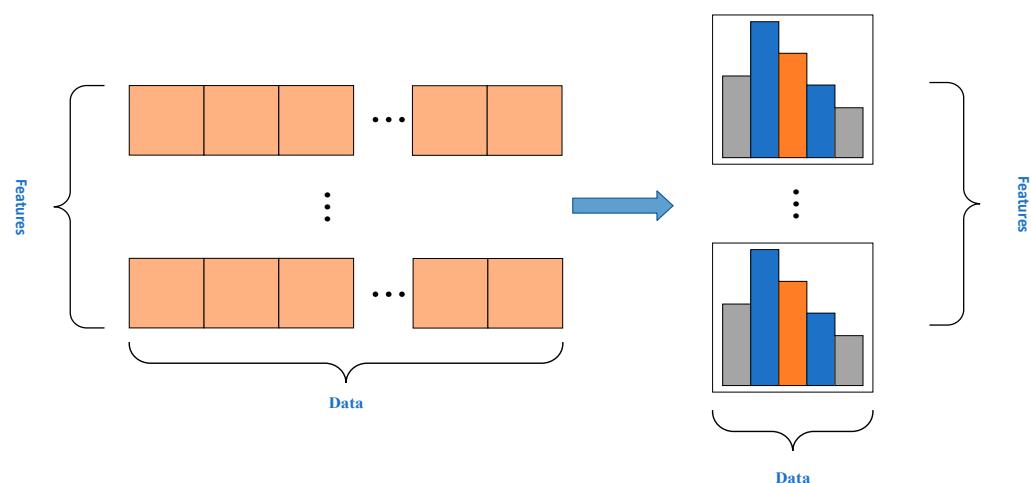
Based on the criterion of minimizing the objective function, XGBoost uses the Taylor second-order expansion term to approximate instead of the objective function, and treats the objective function as a quadratic function of the variable  $\omega_j$ , and then finds the optimal variable  $\omega_j^*$  and the optimal value of the objective function  $B_{Obj}$ .

$$\omega_j^* = -\frac{G_j}{H_j + \lambda}\tag{11}$$

$$B_{Obj}^* = -\frac{1}{2} \sum_{j=1}^T \frac{G_j}{H_j + \lambda} + \gamma T\tag{12}$$

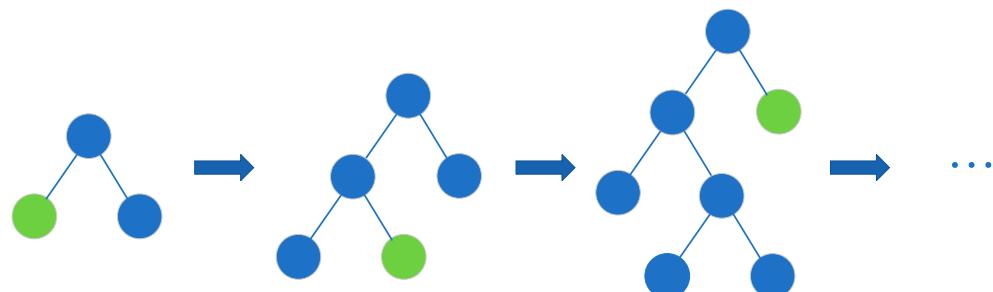
## 2. LightGBM

LightGBM is similar to XGBoost in that the negative gradient of the loss function is used as an approximation of the residuals of the current decision tree to fit the new decision tree [25]. At the same time, LightGBM is also optimized on this basis, and the decision tree algorithm of the histogram is selected. The basic idea is to box the eigenvalues, discretize the continuous floating-point eigenvalues into  $k$  integers to form a box, and construct a histogram with a width of  $k$ . After that, the data are traversed, and the statistics are accumulated in the histogram by the discrete values as the index, and then the optimal segmentation point is found by the discrete values obtained from the histogram shown in Figure 1.



**Figure 1.** Schematic diagram of the LightGBM algorithm.

Since the histogram algorithm does not need to consume additional storage resources to save the pre-sorted results, only the discretized values are required. Thus, LightGBM can effectively reduce memory usage. Additionally, because there is no need to traverse the original feature dataset during operations; instead, calculations are performed based on the constructed histogram just  $k$  times. This reduces the complexity of the calculation and improves computational efficiency. Building upon the histogram algorithm, a leaf-wise growth strategy, as illustrated in Figure 2, is adopted. Nodes with the largest splitting gain are preferentially selected for splitting, followed by selective treatment of leaf nodes within the same layer, which further reduces computational overhead.

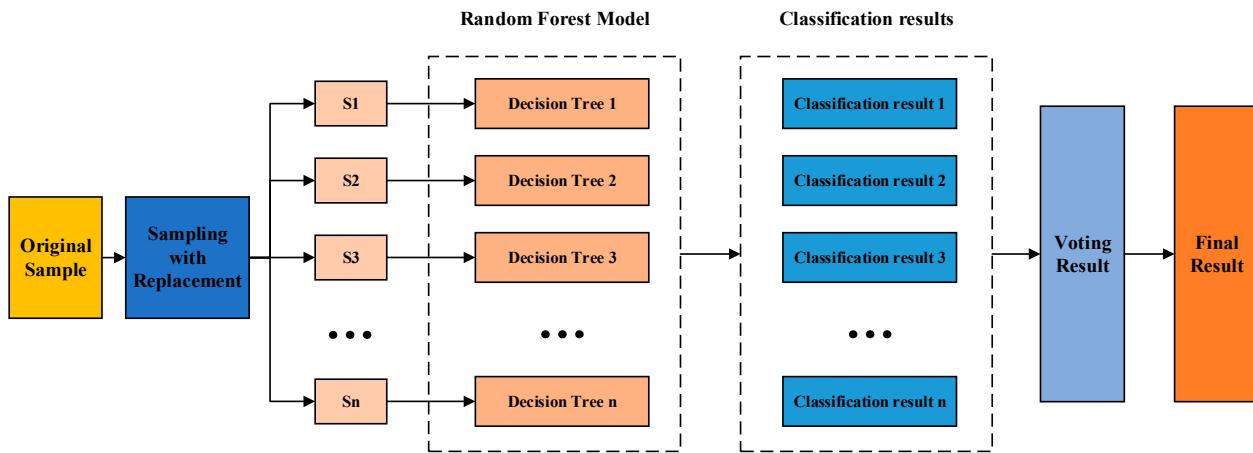


**Figure 2.** Schematic diagram of the Leaf-Wise algorithm.

### 3. Random forest

The random forest algorithm is an ensemble classifier developed from the regression decision tree proposed by Breiman, which uses the bootstrap method (Bootstrap) resampling technology to randomly select a certain number of samples from the original training sample set to generate a new training sample set. It then uses these self-service sample sets to construct multiple classification trees to form a random forest. For the new data sample, the random forest determines the reliability of the classification results based on the voting of the classification tree. Random forests essentially work by integrating multiple decision trees, each of which is built based on independently drawn samples [26]. The process of the random forest algorithm is shown in Figure 3.

Although the classification ability of a single tree may be relatively weak, by generating a large number of decision trees, the classification results of each tree are counted against one test sample, so as to select the most likely classification results. Therefore, the random forest algorithm has good convergence and the ability to prevent overfitting, and the number of classification trees in this study is set to 100.



**Figure 3.** Random forest algorithm flow.

#### 4. AdaBoost

Boosting is a sequential ensemble learning strategy involving the iterative addition of many models (weak learners or base estimators) to improve the overall performance of the model. In AdaBoost, during each iteration, the weights of misclassified samples are continuously increased while the weights of correctly classified samples are decreased, progressively improving the classification accuracy of misclassified samples [27]. Ultimately, the weak classification models obtained from each iteration are combined through weighted summation to derive the final ensemble classification model. The steps of the AdaBoost algorithm are as follows.

Step 1: Initialization. Let the iteration count  $t = 1$ , and denote the training set as  $D$ ; initialize the weights  $D_t(i)$  of each sample as follows:

$$D_t(i) = \frac{1}{L}, (i = 1, 2, 3 \dots L) \quad (13)$$

Step 2: Train the weak classifier. Input  $D$  into the learning algorithm to obtain the weak classifier  $h_t(x)$  for the current iteration, where the weights of each sample in  $D$  are  $D_t(i), (i = 1, 2, 3 \dots L)$ .

Step 3: Update the sample weights. Firstly, the error rate  $\varepsilon_t$  of the weak classification model  $h_t(x)$  in the weight distribution  $D_t(i), (i = 1, 2, 3 \dots L)$  is calculated.

$$\varepsilon_t = \sum_{i=1}^L D(i) \cdot l(h_t(x_i) \neq y_i) \quad (14)$$

where  $l(h_t(x_i) \neq y_i)$  takes a value of 1 when it occurs in  $h_t(x_i) \neq y_i$ , and 0 in all other cases. Next, based on the error rate  $\varepsilon_t$ , calculate the weight  $\alpha_t$  of the weak classifier  $h_t(x)$ .

$$\alpha_t = l_r \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right) \quad (15)$$

where  $l_r$  is the learning rate of the weak classification model for each iteration. Based on the weights of the weak classification model, the weights of each sample in the training set are updated:

$$D_{t+1}(i) = \frac{D(i) \cdot e^{-\alpha_t y_t h_t(x_i)}}{\sum_{i=1}^N D_t(i) \cdot e^{-\alpha_t y_t h_t(x_i)}}, (i = 1, 2, \dots, L) \quad (16)$$

Step 4: Output a strong classification model. If the current number of iterations is  $t < T$ , return to step 2. Otherwise, multiple weak classification models are weighted to obtain the final ensemble classification model  $H(x)$ .

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (17)$$

In this study, Decision Tree (DT) was selected as the weak classification model, and the DT model was selected to use the Gini index as the tree node selection index.

$$Gini = 1 - \sum_{m=1}^M p_{jm}^2 \quad (18)$$

Among these,  $M$  represents the number of class labels in the dataset, and  $p_{jm}$  denotes the proportion of class  $m$  in node  $j$ . A smaller Gini index indicates less uncertainty, reflecting a better classification outcome; therefore, the feature that minimizes the Gini index after splitting is chosen as the tree node.

### 5. Decision tree

The decision tree is a machine learning model for classification and regression that makes predictions by building a tree-like structure. The root node of the tree represents the entire data set, and the inner nodes split the data based on characteristics until it reaches the leaf nodes, which provide the final prediction. The decision tree selects the optimal features for division during training to maximize information gain or reduce mean square error. Its advantages include ease of understanding and interpretation and no need for feature scaling, but it is also prone to overfitting and sensitive to small changes. To prevent overfitting, the decision tree is often pruned, and some branches are removed to improve the generalization ability of the model [28]. The process of building it is as follows.

- (1) Feature selection: The optimal feature is selected through some criteria (such as information gain, Gini coefficient, etc.).
- (2) Split nodes: Splits the data into different subsets based on the selected characteristics.
- (3) Recursive construction: The process of feature selection and splitting is repeated for each subset until the stop condition is met.
- (4) Pruning: After the tree construction is completed, pruning may be performed to reduce overfitting and improve the generalization ability of the model.

### 6. SVM:

Support Vector Machine (SVM) is a supervised learning method that is mainly used to classify problems. The main idea of the SVM is to find an optimal hyperplane that maximizes the spacing between the different classes [29]. The core of SVM classification is the optimal classification hyperplane theory, which refers to the classification surface that can accurately separate two types of data and have the largest gap. The training of the support vector machine is realized by constructing an optimization problem, introducing the Lagrange multiplier to convert the constraints into objective functions, and using the dual problem to solve the optimal solution. In the case of two-dimensional space, there is a hyperplane.

$$w \cdot x + b = y \quad (19)$$

where  $x$  is the input sample;  $w$  is the weight vector of the support vector machine;  $b$  is the offset.

The hyperplane can separate different classes with the maximum margin. Thus, the support vector machine (SVM) converts the aforementioned classification problem into the following function optimization problem, namely:

$$\underset{w}{\text{minimize}} \frac{\|w\|^2}{2} \quad (20)$$

The constraints are

$$y_i(w \cdot x_i + b) \geq 1, \forall i \quad (21)$$

where  $x$  is the input sample;  $w$  is the weight vector of the support vector machine.

By introducing the Lagrange multiplier, the above optimization problem can be transformed into a dual problem, in which our objective function becomes

$$\text{maximize } L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j x_i x_j \quad (22)$$

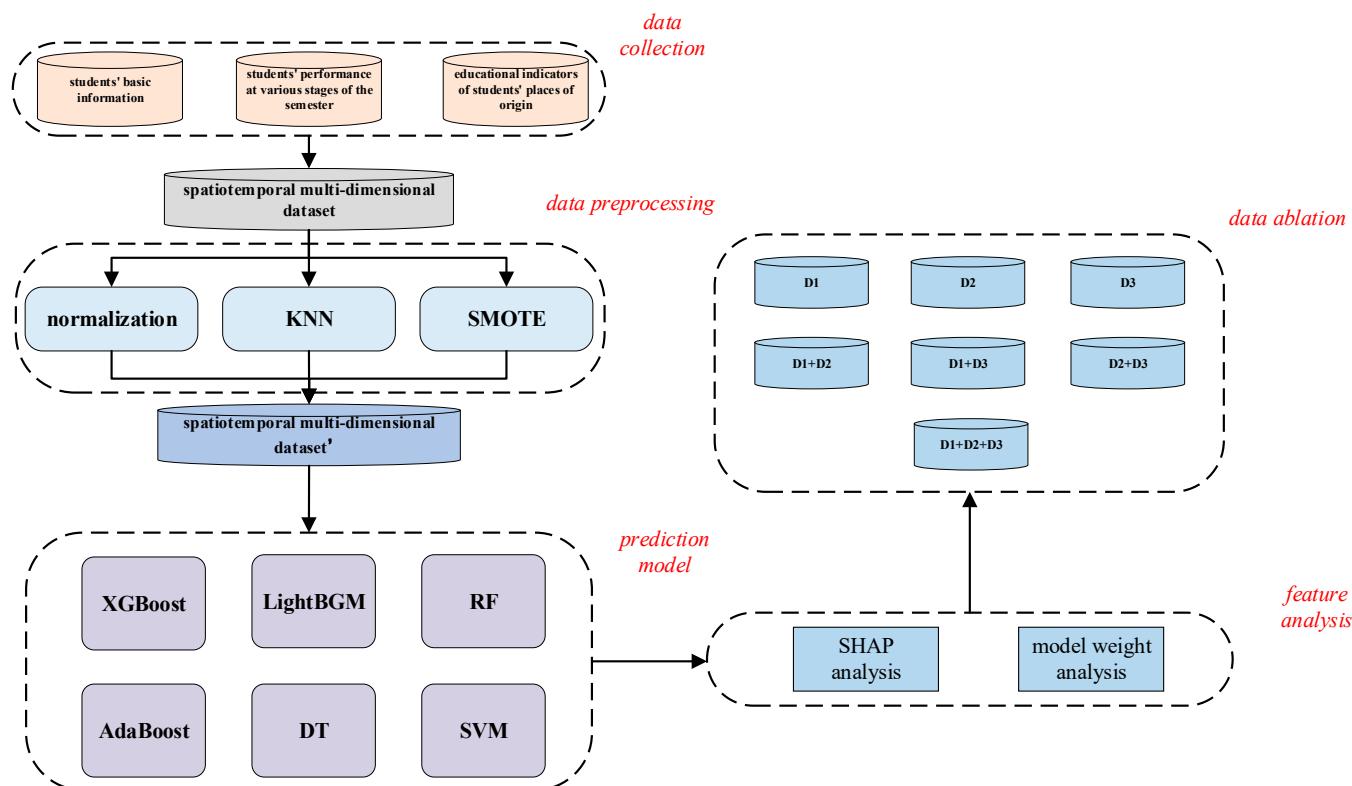
where  $\alpha_i$  is the Lagrange multiplier.

After solving this optimization problem, the decision boundary can be obtained, and the final classifier can be represented by the weight of the support vector. For linearly indivisible data, SVM can map the data to a high-dimensional space through a kernel function in order to find a linearly separable hyperplane.

### 3.1.2. SHAP Analysis

Shapley Additive Explanations (SHAP) serves to enhance the explainability and transparency of machine learning models [30]. Grounded in Shapley values from game theory, this technique aims to measure each feature's contribution to the predictions made by the model. Possessing traits of fairness, consistency, and local interpretability, SHAP evaluates each feature's impact across all potential combinations. This ensures no feature's influence gets underestimated or overestimated due to interactions with others. Should a feature demonstrate consistent significance across varying models, SHAP reflects this consistently in its attributions. Additionally, SHAP provides explanations tailored to individual prediction outcomes, aiding comprehension of specific prediction rationales [31].

The specific steps are as follows (Figure 4).



**Figure 4.** Overall design drawing.

1. Data Collection and Integration: Collect three types of data, including students' demographic information, scores at various stages of the semester, and educational indicators from their places of origin, and integrate these three types of data.
2. Data Preprocessing: Impute missing data using the KNN interpolation method and handle imbalanced data using SMOTE.

3. Training Machine Learning Models: Obtain a multidimensional spatiotemporal dataset through steps (1) and (2), and use this dataset to train six machine learning models, including XGBoost, LightGBM, Random Forest, AdaBoost, Decision Tree, and SVM.
4. Optimal Model Selection: Evaluate the models using four metrics—accuracy, recall, precision, and F1 score—to select the best predictive model.
5. Feature Importance Analysis: Perform SHAP analysis and weight analysis on the models to assess the importance of each feature.
6. Data Ablation: Combine and divide the multidimensional spatiotemporal dataset into seven sub-datasets, train the machine learning models using these seven sub-datasets, and analyze the experimental results.

#### 4. Experimental Results

##### 4.1. Forecast Results

In this study, the students' grades (G) were divided into five levels (Mark G < 60 as 0,  $60 \leq G < 70$  as 1,  $70 \leq G < 80$  as 2,  $80 \leq G < 90$  as 3, and  $90 \leq G \leq 100$  as 4). Machine learning models, such as XGBoost, LightGBM, Random Forest, AdaBoost, Decision Tree and SVM, were selected to predict students' academic performance. Four model evaluation indexes (accuracy, recall, precision, and F1 value) were used to evaluate each model [32]. The hyperparameter settings of the experimental models are described in Table 2, and the testing equipment and software are described in Table 3.

**Table 2.** Model hyperparameter settings.

Models	Hyperparameter Settings
XGBoost	n_estimators = 100, max_depth = 6, learning_rate = 0.3, subsample = 1.0, colsample_bytree = 1.0, min_child_weight = 1, gamma = 0, reg_alpha = 0, reg_lambda = 1, scale_pos_weight = 1
LightGBM	boosting_type = 'gbdt', num_leaves = 31, max_depth = -1, learning_rate = 0.1, n_estimators = 100, subsample_for_bin = 200,000, min_split_gain = 0.0, min_child_weight = 0.001, min_child_samples = 20, subsample = 1.0, subsample_freq = 0, colsample_bytree = 1.0, reg_alpha = 0.0, reg_lambda = 0.0, class_weight = None, importance_type = 'split', n_jobs = -1, silent = True, random_state = None
Random Forest	n_estimators = 100, criterion = 'gini', max_depth = None, min_samples_split = 2, min_samples_leaf = 1, min_weight_fraction_leaf = 0.0, max_features = 'auto', max_leaf_nodes = None, min_impurity_decrease = 0.0, bootstrap = True, oob_score = False, warm_start = False
AdaBoost	base_estimator = DecisionTreeClassifier(max_depth = 1), n_estimators = 50, learning_rate = 1.0, algorithm = 'SAMME.R', random_state = None
SVM	C = 1.0, Kernel = 'rbf', Degree = 3, Gamma = 'scale', coef0 = 0.0, Shrinking = True, Probability = False, tol = 0.001, cache_size = 200, class_weight = None, Verbose = False, max_iter = -1, decision_function_shape = 'ovr', break_ties = False, random_state = None
Decision Tree	Criterion = 'gini', splitter = 'best', max_depth = None, min_samples_split = 2, min_samples_leaf = 1, min_weight_fraction_leaf = 0.0, max_features = None, random_state = None, max_leaf_nodes = None, min_impurity_decrease = 0.0, class_weight = None, Presort = 'deprecated'

**Table 3.** Testing equipment and software.

Equipment and Software	Equipment Model and Software Version
CPU	13th Gen Intel(R) Core(TM) i5-13500HX 2.50 GHz
GPU	NVIDIA GeForce RTX 4060
Operating system	CentOS 7.6
Testing software version	python 3.9, numpy 1.23.3, pandas 1.5.0, scikit-learn 1.1.2

##### 1. Accuracy

*Accuracy* serves as a frequent metric for assessing a classification model's performance. It represents the ratio of correctly predicted samples to the total sample count. The calculation formula for the accuracy rate is presented as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (23)$$

The metric  $TP$ , standing for True Positive, signifies the quantity of positive class instances accurately identified by the model. For  $TN$ , known as True Negative, it denotes the quantity of negative class instances correctly recognized.  $FP$ , False Positive, signifies the count of negative class instances misclassified by the model as positive, typically termed a “false alarm”.  $FN$ , or False Negative, involves cases where the model erroneously classifies positive class instances as belonging to the negative category, commonly called “false negatives”.

## 2. Recall

*Recall*, an important evaluation metric for classification models, particularly in the context of unbalanced datasets, measures the model’s effectiveness in identifying all positive class examples correctly.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (24)$$

$TP$ , or True Positive, represents the count of true positives, while  $FN$ , or False Negative, indicates the number of false negatives.

## 3. F1 Score

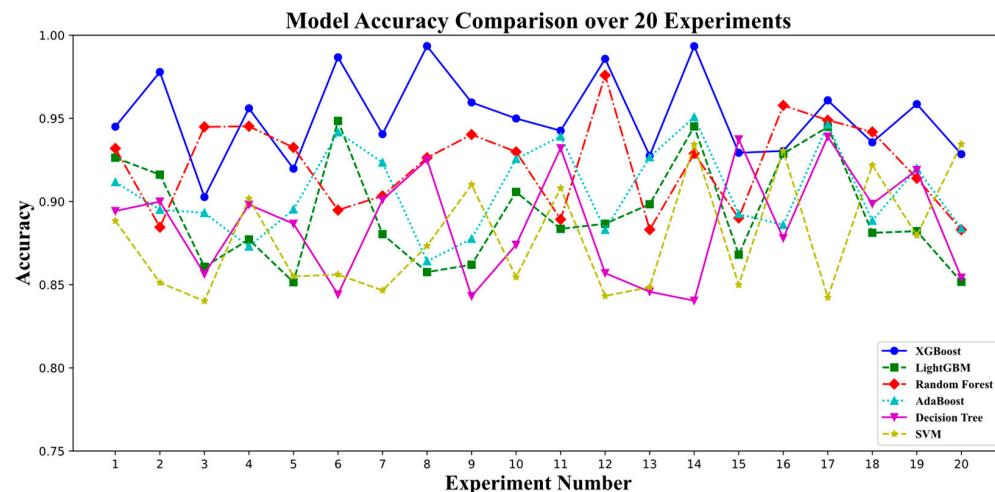
The  $F1$  score, an important indicator for assessing classification models, particularly in cases of unbalanced dataset categories, is extensively utilized in machine learning and deep learning areas. This score combines precision and recall. The  $F1$  Score represents the harmonic mean of precision and recall, and its formula is as follows:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (25)$$

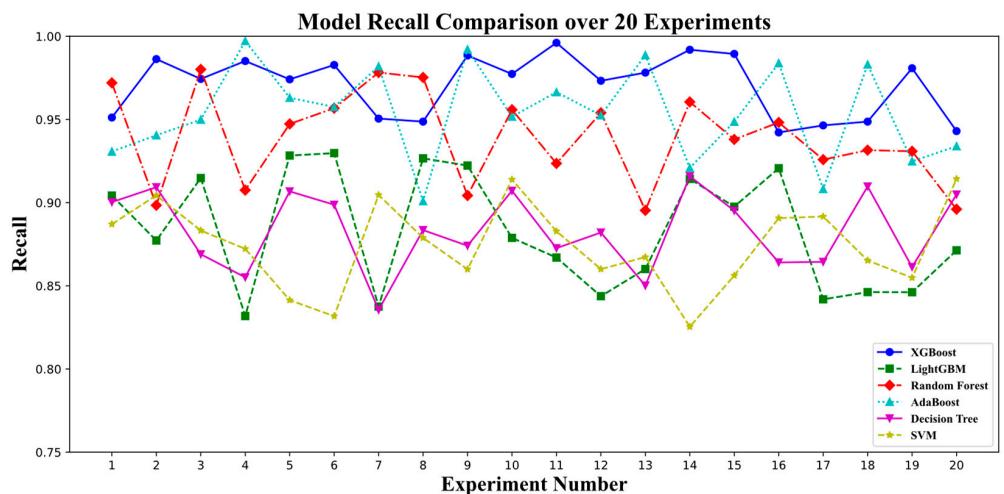
The  $F1$  score is an important performance evaluation metric and is a harmonic average of precision and recall.

In this study, the holdout method facilitated 20 rounds of comparative experiments for each model. For every experimental round, 70% of the entire dataset were randomly chosen as the training subset, with 30% forming the test subset. Across the 20 rounds, accuracy, recall, precision, and  $F1$  scores were computed for each model to assess performance.

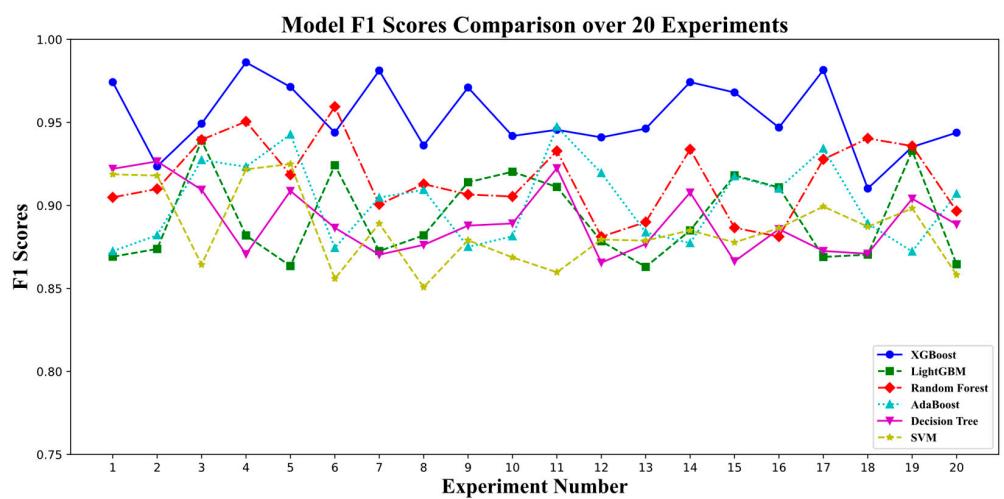
As shown in Figures 5–8, each model showed excellent prediction results on three types of feature datasets, with XGBoost performing best on all four evaluation metrics (accuracy, recall, precisions, and  $F1$  scores).



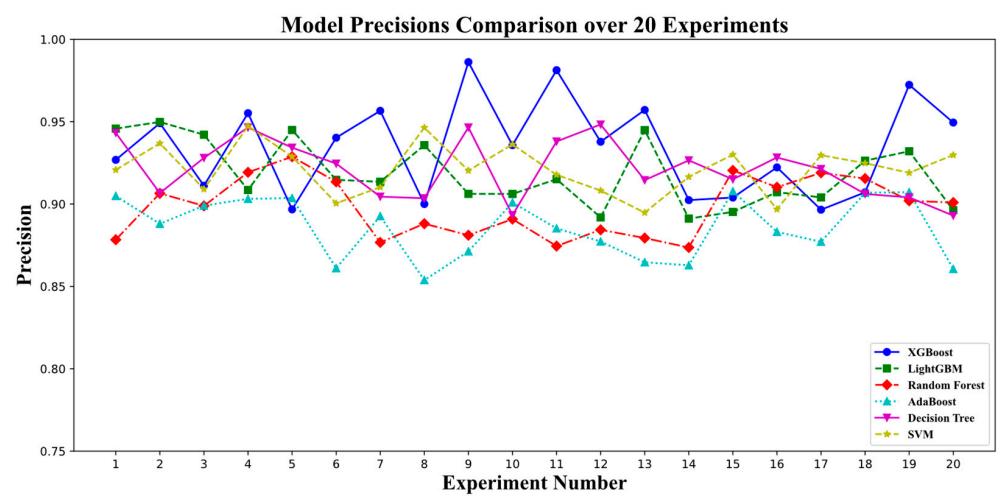
**Figure 5.** The accuracy of each model on the dataset with three types of features.



**Figure 6.** The recall of each model on the dataset with three types of features.



**Figure 7.** The precisions of each model on the dataset with three types of features.



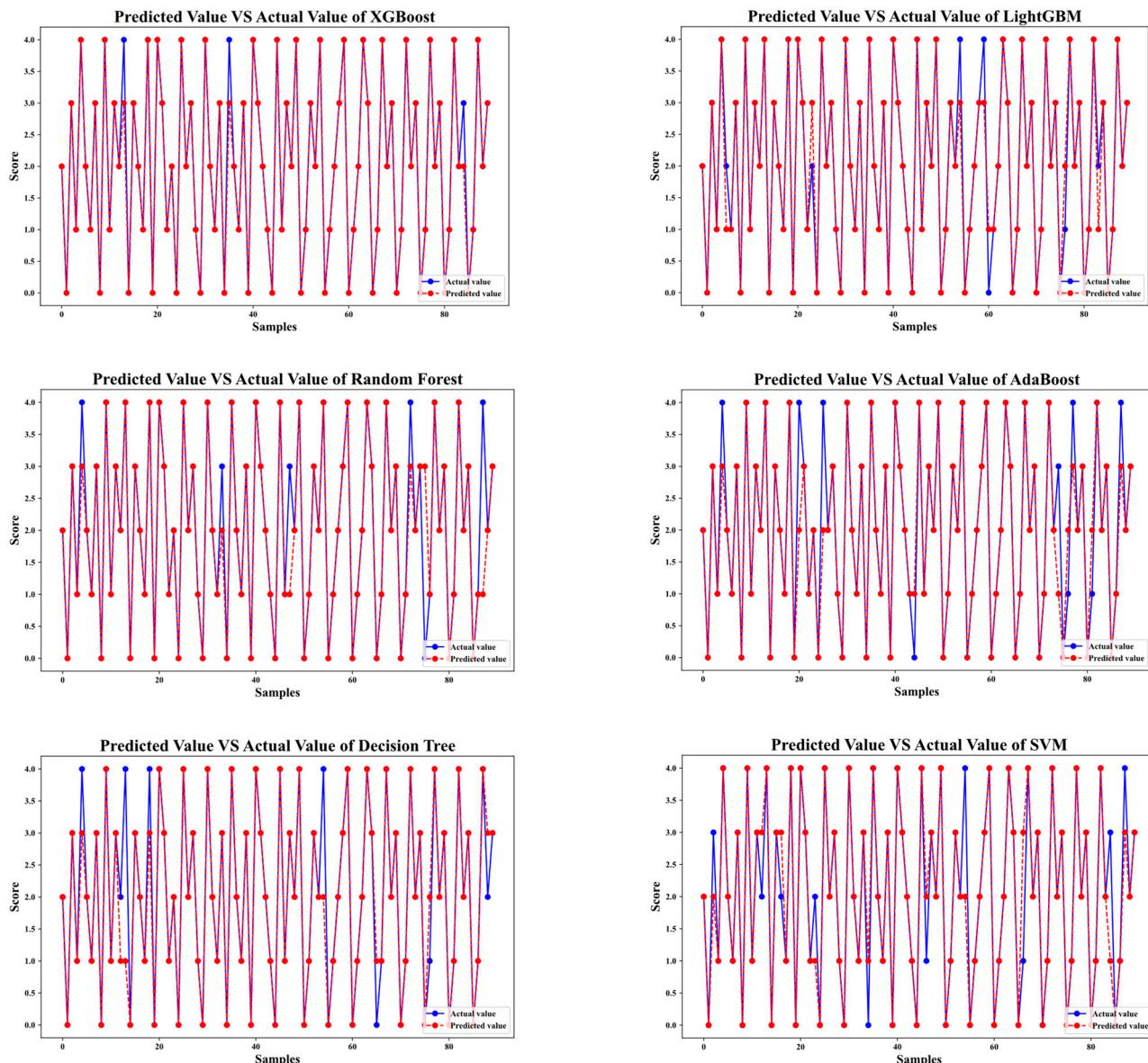
**Figure 8.** The F1 scores of each model on the dataset with three types of features.

In this study, the average values of the four evaluation indicators of the 20 rounds of comparative trials of each model were counted, as shown in Table 4.

**Table 4.** The accuracy, recall, precisions, and F1 scores of different algorithms.

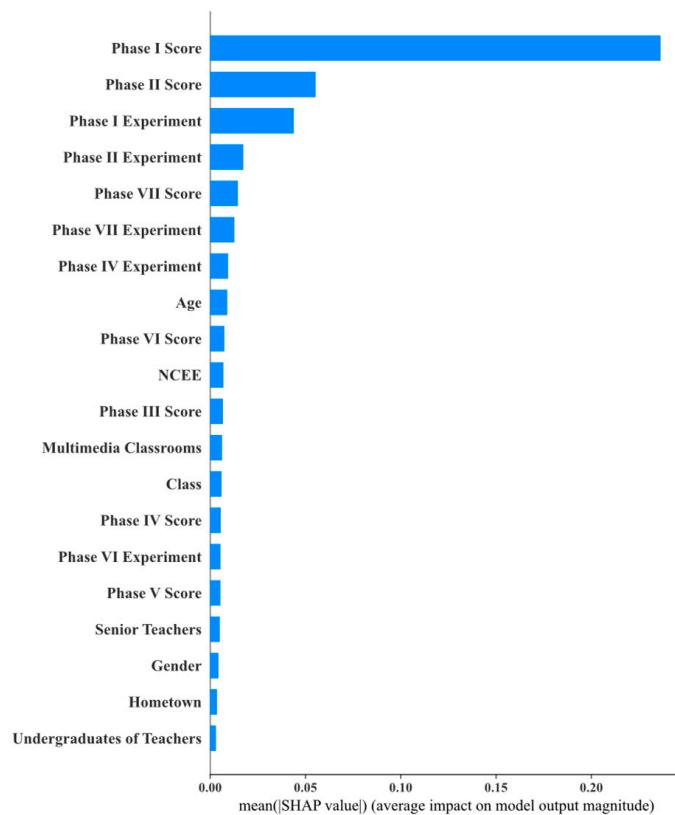
Model	XGBoost	LightBGM	RF	AdaBoost	DT	SVM
Accuracy	0.95	0.90	0.92	0.89	0.87	0.89
Recall	0.96	0.85	0.93	0.92	0.85	0.88
Precision	0.93	0.93	0.91	0.87	0.92	0.93
F1 score	0.94	0.89	0.92	0.89	0.88	0.90

Figure 9 shows the comparison of the true and predicted values of each model.

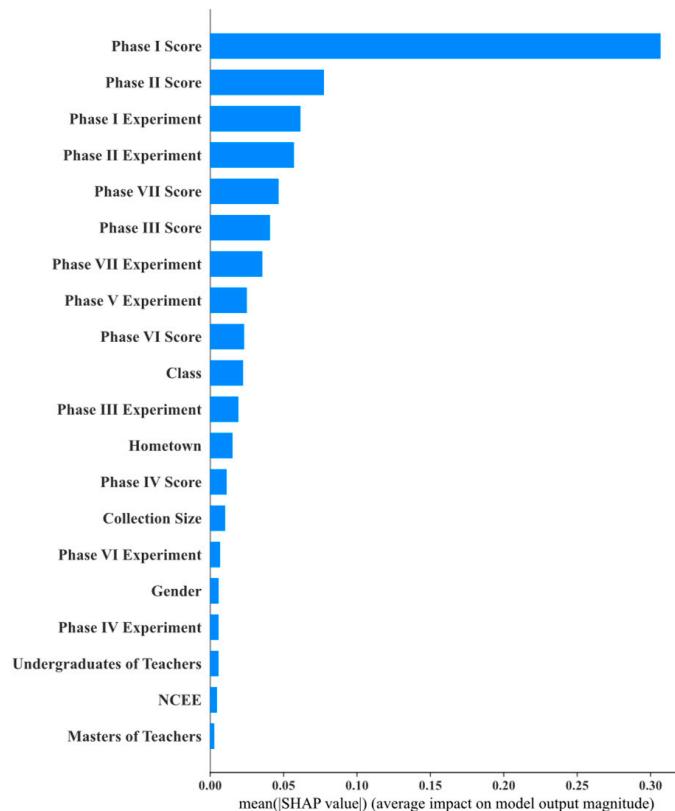
**Figure 9.** Comparison of the true and predicted values of each model.

#### 4.2. Feature Analysis

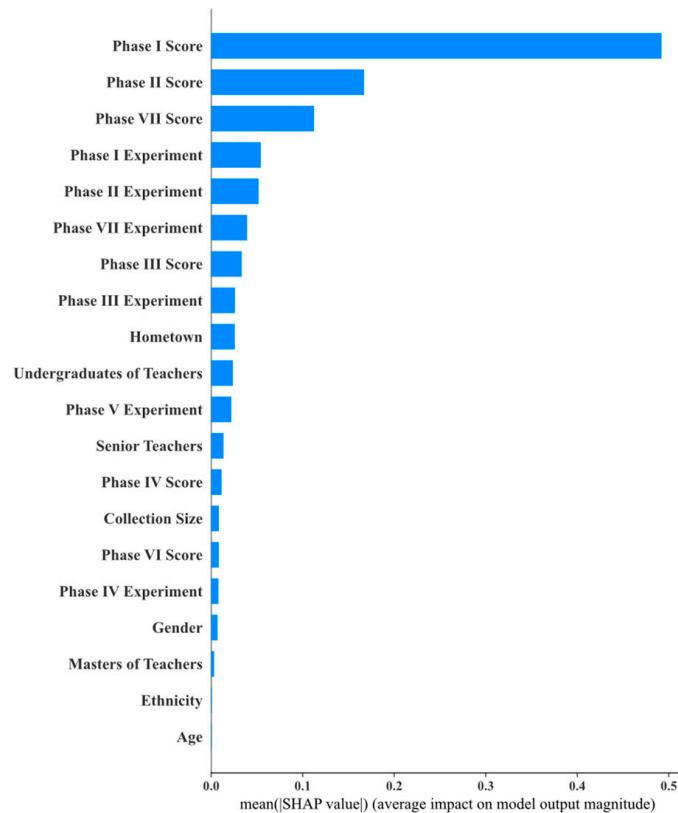
In this study, we applied the SHAP (Shapley Additive Explanations) method to analyze the feature importance of six machine linear models to understand the contribution of each feature to the prediction results of the model. SHAP values provide a consistent way to quantify the impact of each feature on a single prediction, revealing the model's decision-making process. Figures 10–15 show the average SHAP ranking of each feature of the six models.



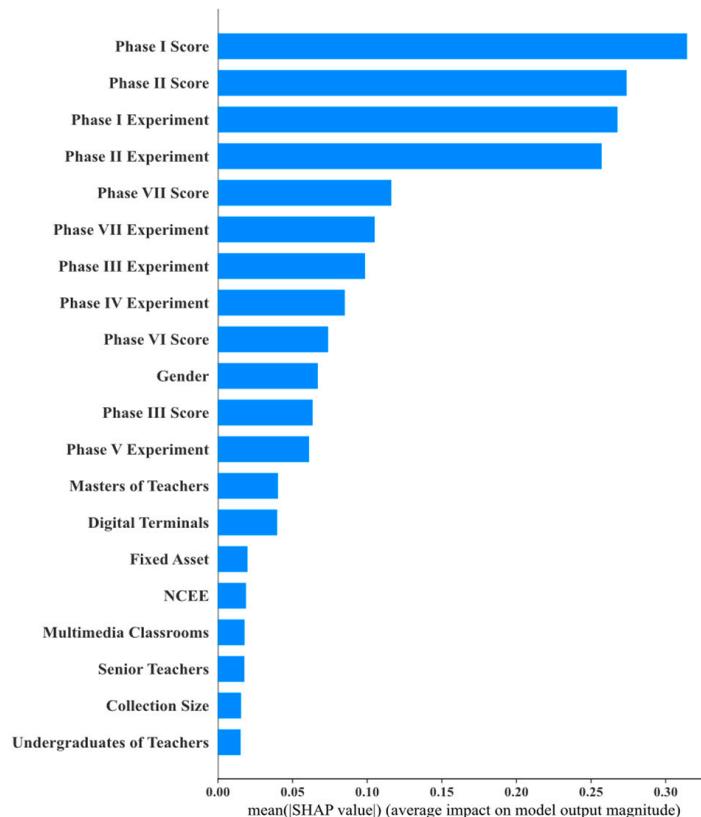
**Figure 10.** XGBoost.



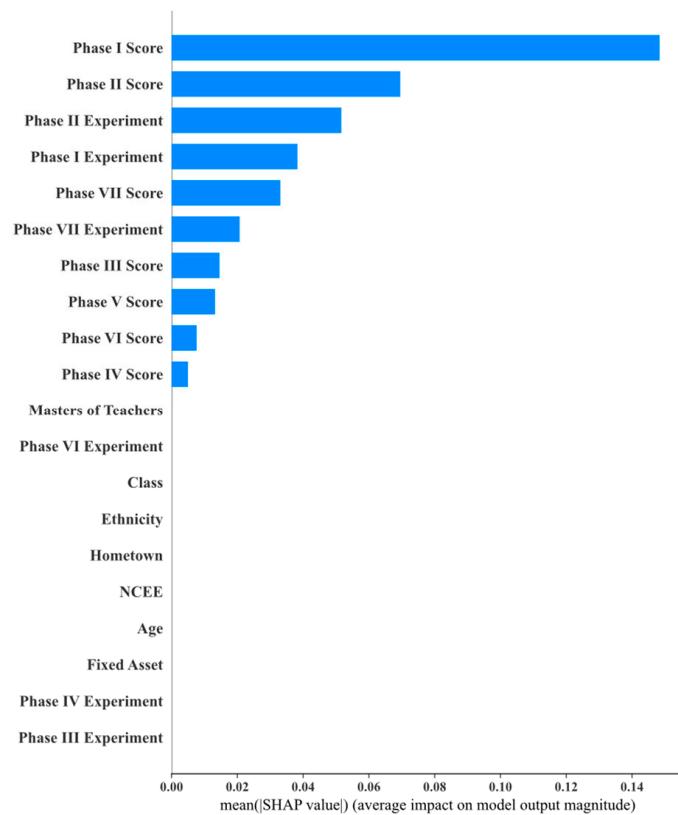
**Figure 11.** Random Forest.



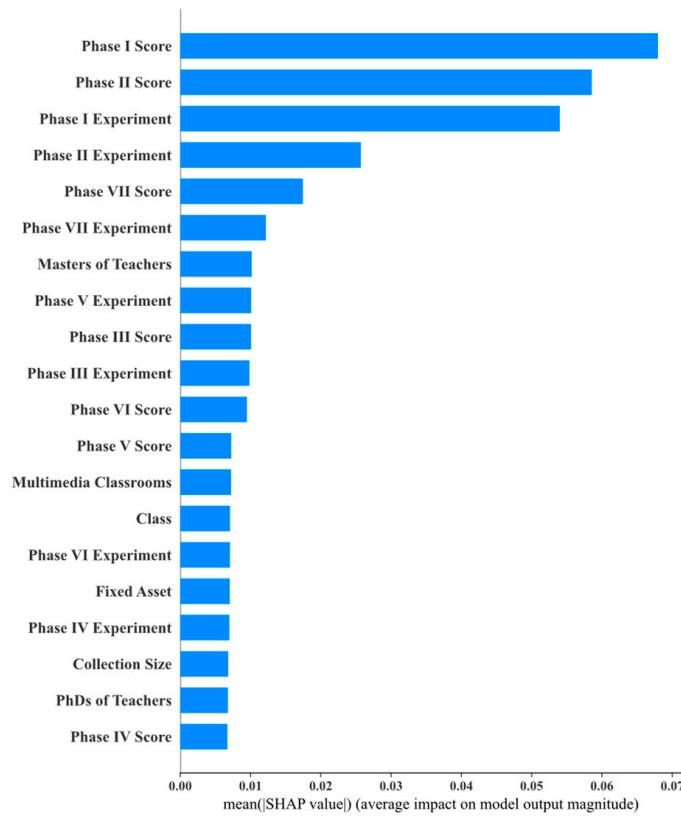
**Figure 12.** LightGBM.



**Figure 13.** AdaBoost.



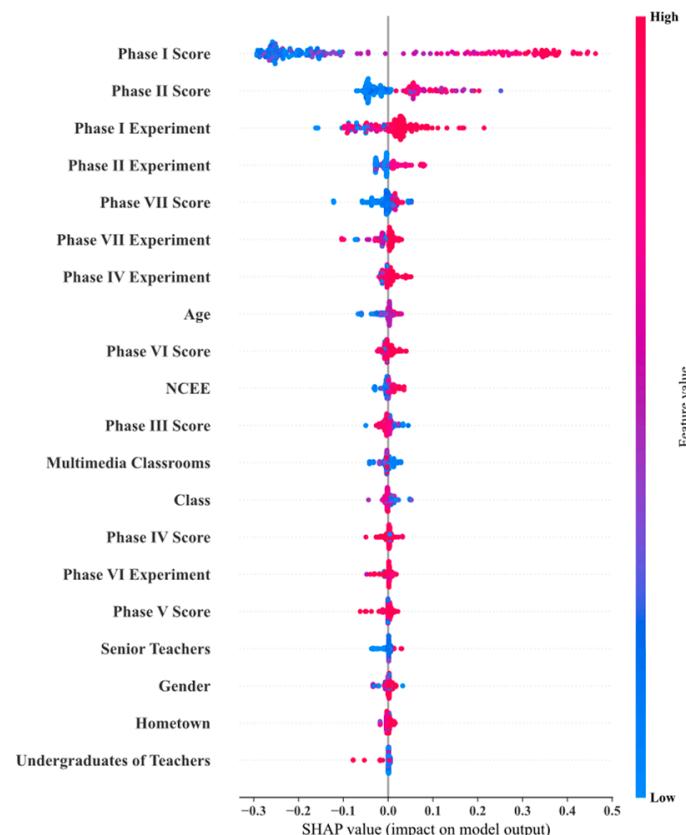
**Figure 14.** Decision Tree.



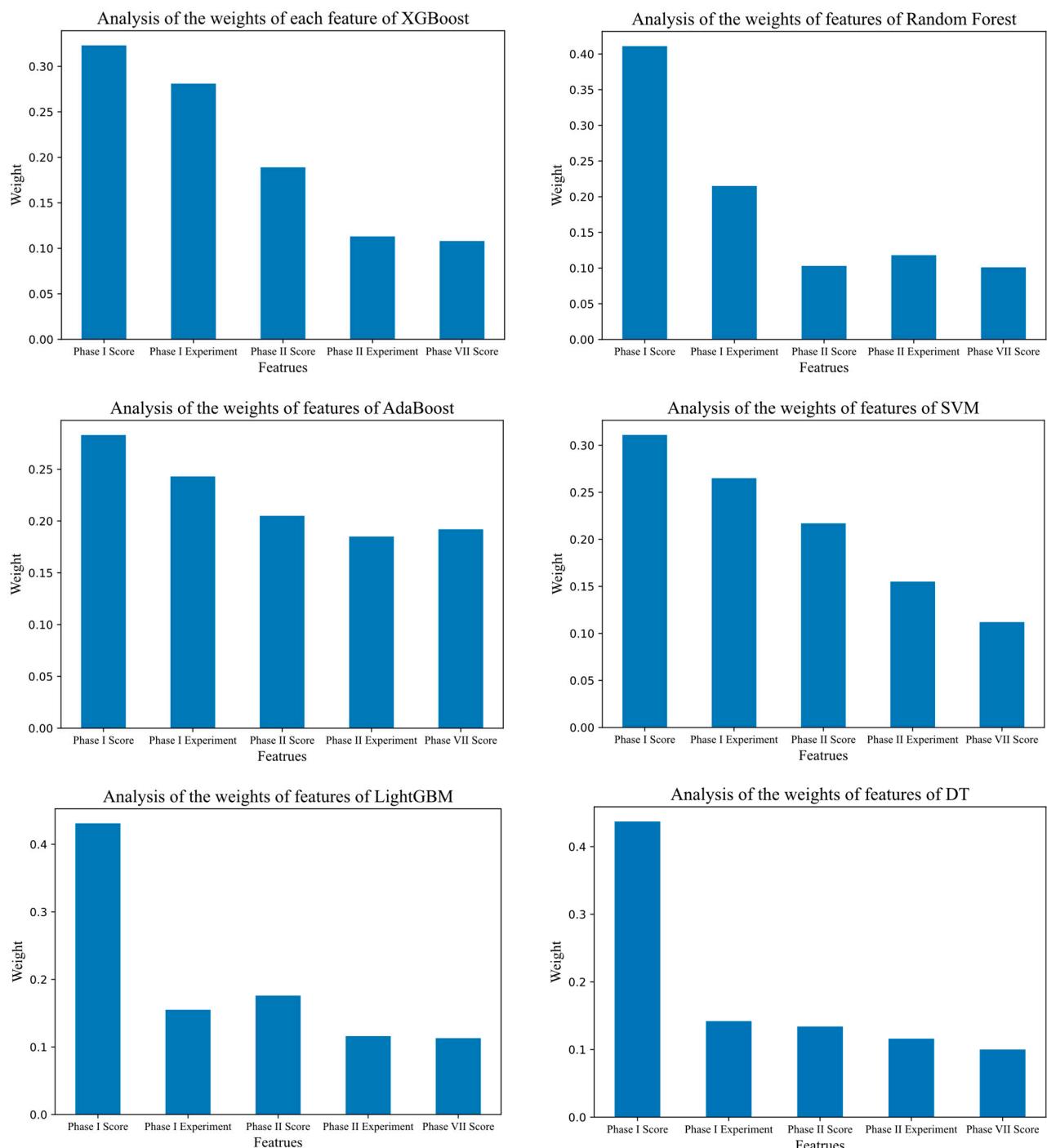
**Figure 15.** SVM.

As shown in Figures 10–16, this study used SHAP analysis to analyze the degree to which each attribute contributed to the prediction results. We found that students' homework scores and experiment scores had the most significant impact on outcomes in Stages 1, 2, and 7. In this study, we analyzed the weights of each model to identify the features that had a significant impact on the prediction results. We selected the top five features for SHAP analysis, and the weights for each model are shown in Figure 17.

This study identified the top five features that had the greatest impact on the prediction results through an analysis of the weights of various models. The top five features ranked by XGBoost weight values are as follows: Phase I Score = 0.323; Phase I Experiment = 0.281; Phase II Score = 0.189; Phase II Experiment = 0.113; Phase VII Score = 0.108. The top five features ranked by Random Forest weight values are: Phase I Score = 0.431; Phase I Experiment = 0.155; Phase II Score = 0.176; Phase II Experiment = 0.116; Phase VII Score = 0.113. The top five features ranked by LightGBM weight values are as follows: Phase I Score = 0.411; Phase I Experiment = 0.215; Phase II Score = 0.103; Phase II Experiment = 0.118; Phase VII Score = 0.101. The top five features ranked by LCE weight values are as follows: Phase I Score = 0.283; Phase I Experiment = 0.243; Phase II Score = 0.205; Phase II Experiment = 0.185; Phase VII Score = 0.192. The top five features ranked by Decision Tree weight values are as follows: Phase I Score = 0.437; Phase I Experiment = 0.142; Phase II Score = 0.134; Phase II Experiment = 0.116; Phase VII Score = 0.110. The top five features ranked by SVM weight values are as follows: Phase I Score = 0.311; Phase I Experiment = 0.265; Phase II Score = 0.217; Phase II Experiment = 0.155; Phase VII Score = 0.112.



**Figure 16.** XGBoost Summary SHAP Plot.



**Figure 17.** The weights for each model.

#### 4.3. Data Ablation

In this study, ablation experiments were performed on the dataset. In the data preparation stage of the study, we divided the dataset into three categories: student basic information data, student performance data at each stage, and student origin education indicator data, which we recorded as D1, D2, and D3. We divided and combined the data into six datasets: D1, D2, D3, D1 + D2, D2 + D3, D1 + D3, D1 + D2 + D3, and we used these six datasets to train the six machine learning models above, according to the 70% training set and 30% test set. Then, we evaluated the results with four evaluation indicators: accuracy, precision, recall, and F1 value, as shown in Table 5.

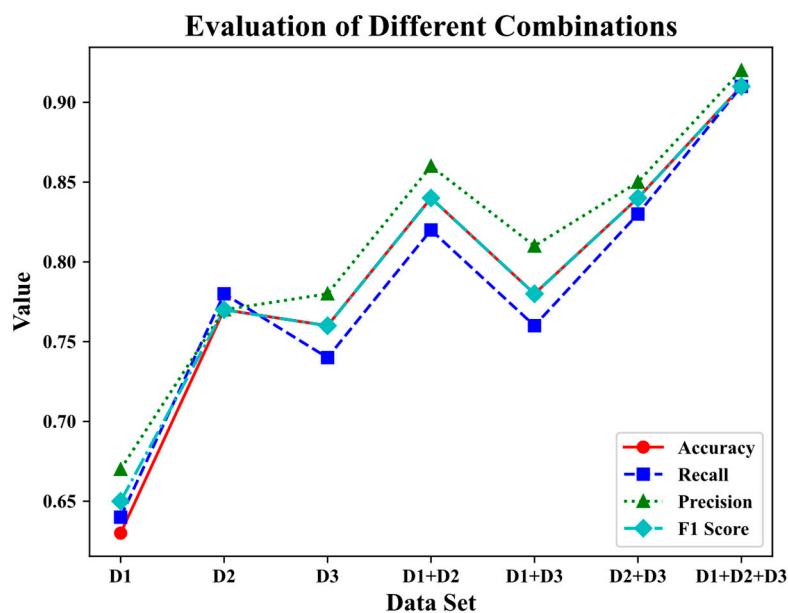
**Table 5.** Result of ablation experiments.

Dataset	Model	ACC	RC	PC	F1
D1	XGB	0.68	0.70	0.67	0.68
	LGBM	0.62	0.61	0.65	0.63
	RF	0.64	0.58	0.72	0.64
	AB	0.63	0.70	0.60	0.65
	DT	0.60	0.61	0.73	0.66
	SVM	0.61	0.62	0.65	0.63
D2	XGB	0.83	0.80	0.85	0.82
	LGBM	0.79	0.75	0.83	0.79
	RF	0.75	0.82	0.70	0.76
	AB	0.81	0.80	0.83	0.81
	DT	0.71	0.75	0.69	0.72
	SVM	0.74	0.77	0.70	0.73
D3	XGB	0.80	0.81	0.78	0.79
	LGBM	0.76	0.75	0.78	0.76
	RF	0.77	0.76	0.79	0.77
	AB	0.75	0.70	0.80	0.75
	DT	0.73	0.71	0.77	0.74
	SVM	0.72	0.68	0.75	0.71
D1 + D2	XGB	0.87	0.86	0.90	0.88
	LGBM	0.85	0.81	0.88	0.84
	RF	0.82	0.77	0.83	0.80
	AB	0.86	0.85	0.89	0.87
	DT	0.80	0.81	0.79	0.80
	SVM	0.83	0.82	0.85	0.83
D1 + D3	XGB	0.82	0.85	0.80	0.82
	LGBM	0.80	0.76	0.82	0.79
	RF	0.79	0.78	0.83	0.80
	AB	0.76	0.73	0.80	0.76
	DT	0.73	0.71	0.77	0.74
	SVM	0.75	0.73	0.81	0.77
D2 + D3	XGB	0.88	0.86	0.90	0.88
	LGBM	0.85	0.83	0.88	0.85
	RF	0.82	0.77	0.86	0.81
	AB	0.85	0.81	0.90	0.85
	DT	0.81	0.85	0.77	0.81
	SVM	0.83	0.88	0.79	0.83
D1 + D2 + D3	XGB	0.95	0.97	0.94	0.95
	LGBM	0.90	0.88	0.92	0.90
	RF	0.93	0.94	0.90	0.92
	AB	0.91	0.95	0.88	0.91
	DT	0.88	0.86	0.93	0.89
	SVM	0.89	0.87	0.92	0.89

In this study, four indicators were used, including accuracy, recall, precision, and F1 score, to evaluate the prediction results of six machine learning models under seven datasets. From the overall experimental results, training the model with complete features gave the best experimental results. This also proves the rationality and effectiveness of feature selection in the initial stage of this study. At the same time, we found that the use of the D2 dataset (the characteristics of students' performance at each stage of the semester) was the most effective in predicting student achievement.

This study used four metrics—accuracy, recall, precision, and F1 score—to evaluate the prediction results of six machine learning models on seven datasets. Overall, the experimental results indicate that training the models with complete features yields the best outcomes. This also validates the rationality and effectiveness of the feature selection in

the initial stage of this study. The average values of the four metrics for all models across the seven datasets are shown in Figure 18. It shows the performance of each machine learning on different datasets. Additionally, the experimental results demonstrate that using the D2 dataset (students' performance at various stages of the semester) provides the best performance among single-feature datasets for predicting students' academic performance. Using the comprehensive spatiotemporal multidimensional feature dataset for predicting students' academic performance outperforms all sub-datasets in terms of effectiveness. Moreover, the model achieved the best prediction results when the three types of data were fused ( $D1 + D2 + D3$ ), which also verified the rationality and effectiveness of fusing the three types of data.

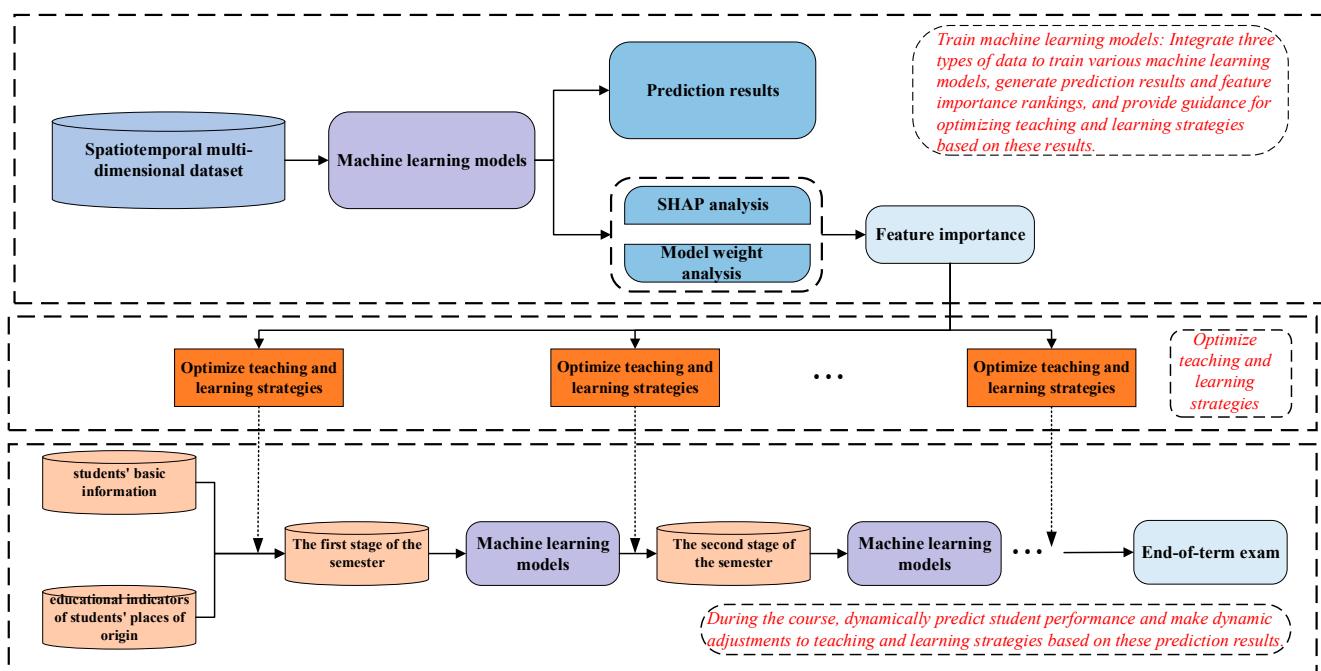


**Figure 18.** The average of the four metrics across all models across seven datasets.

## 5. Applicability and Feasibility

This study provides an applicable and feasible solution for the participants in educational activities, using a dynamically optimized strategy to improve teaching processes, thereby increasing student pass rates and reducing failure rates. The practical application method is shown in Figure 19.

This study involved training machine learning models using historical data to predict the final grades of students. The experimental results demonstrated the effectiveness of the machine learning models when applied to spatiotemporal multi-dimensional feature datasets. Furthermore, through SHAP analysis and weight analysis, the importance ranking of each feature was derived, which holds significant guiding implications for educators in optimizing and adjusting teaching and learning strategies. Educators can allocate resources and attention reasonably based on the importance ranking of features. Adjustments to educational strategies primarily encompass two aspects: students' learning behaviors and teachers' intervention actions. Accordingly, this study proposed specific optimization strategies for both. Additionally, this research paid attention to the temporality of the course progression and the dynamic changes in students' performance at different stages, dynamically adjusting teaching strategies in response to the outcomes of grade predictions at each stage. Moreover, in response to the disparities in educational resources and education quality among students' places of origin, corresponding optimizations to teaching strategies were also made.



**Figure 19.** The operation process of the model in the teaching process.

### 5.1. Guidance for Student Actions

For students, they can utilize this information to improve their learning. Through analysis, it was found that homework and experimental scores significantly impact the final grades during the first, second, and seventh stages of the semester. Therefore, in these stages, students should place special emphasis on the quality of homework completion, treating every assignment task seriously, and promptly summarizing issues encountered in assignments while seeking help from teachers or peers. During the experimental phase, active participation in experimental operations is encouraged, with a focus on methods for data collection and analysis during experiments. If performance in these critical stages is poor, timely adjustments to learning methods should be made, such as increasing study time and enhancing collaborative learning with classmates, to improve performance in subsequent stages. Meanwhile, students can use the model's predictive results to have a preliminary estimate of their potential achievements for the semester, thus setting reasonable learning goals and formulating corresponding study plans. If the predicted grade is low, students should plan review times in advance, addressing knowledge gaps in a targeted manner.

### 5.2. Guidance for Teacher Interventions

For teachers, they can leverage this information to dynamically adjust their teaching strategies. Based on the key stages and important features identified by the predictive model, teachers can optimize their teaching arrangements. During stages that significantly influence final grades, they can increase the depth and breadth of instruction, introduce more practical cases and project-based learning, and foster students' comprehensive application abilities. At the same time, understanding the foundational differences among students according to educational indicators from their regions of origin, teachers can provide additional foundational knowledge tutoring for students coming from areas with relatively weaker educational resources, organizing small preparatory courses or study groups. Furthermore, teachers can use the model's predictive results to promptly identify groups of students facing learning difficulties, offering personalized guidance and support.

For school staff, they can use this information to optimize resource allocation and management decisions. School staff can analyze educational indicators from students' regions of origin to determine the student groups and areas requiring focused support,

allocating educational resources reasonably. Special scholarships or grants can be provided to students from regions with scarce educational resources, encouraging them to actively engage in learning; additional learning resources, such as books and experimental equipment, can be allocated to classes where these students are enrolled. Additionally, the overall teaching effectiveness of the school can be assessed based on the predictive model, leading to the formulation of long-term educational planning and policies. If the predicted grades for a particular subject are generally low, the school will strengthen the faculty team for that subject, organize teacher training, recruit outstanding teachers, and adjust course settings, optimizing course content and syllabi to enhance teaching quality and student learning outcomes.

## 6. Summary and Discussion

In the initial phase, this study gathered a dataset combining three types of features: student personal basic information, student performance at various stages of the semester, and educational indicators from the students' regions of origin. Integrating these three types of features, this study constructed an educational dataset featuring multidimensional spatiotemporal characteristics. The primary challenges during the dataset construction phase mainly centered around the collection of educational data. Due to the inherent privacy of educational data and the lack of deliberate retention of historical educational data by teachers in traditional courses, the collection of educational data proved to be a challenging process in this study. Regarding students' personal information, this type of data has strong privacy concerns. Through active communication with relevant departments at the school and the fulfillment of corresponding confidentiality agreements, the author managed to collect this type of data. Concerning the data on students' performance at different stages, due to the varying designs of courses by teachers, some courses did not include staged assessments of student performance, making it impossible to collect this type of data. (The author also strongly recommends that teachers consider adopting a phased assessment strategy in course design.) Thus, the author surveyed the course designs of multiple courses at the institution and ultimately collected this type of data from a foundational course in the Computer Science major—C Programming. This process required a substantial investment of time and communication efforts, ultimately leading to the construction of an educational dataset featuring multidimensional spatiotemporal characteristics, with all experimental data being real.

Secondly, during the data collection process, we discovered that educational data often suffers from missing values and class imbalance issues. To address these, this study employed KNN interpolation and SMOTE oversampling for preprocessing the dataset, resulting in the final multidimensional spatiotemporal feature dataset for predicting student grades. Subsequently, six machine learning models were selected for this study: XGBoost, Random Forest, LightGBM, AdaBoost, Decision Tree, and SVM. Using the constructed dataset, which was divided into a 70% training set and a 30% testing set, these six machine learning models were trained. A total of 20 rounds of comparative experiments were conducted, dividing the dataset randomly into training and testing sets at a ratio of 7:3, evaluating the six models using accuracy, recall, precision, and F1 score. The experimental results indicated that all six models achieved good performance, with XGBoost showing the best performance. This demonstrates the effectiveness of integrating multidimensional spatiotemporal features in predicting student grades, proving that the comprehensive integration of these features for grade prediction is valid. Following this, SHAP analysis and weight analysis were performed on the six machine learning models to evaluate the importance of different features. The experiments revealed that the scores from homework and experiments in the first, second, and seventh stages of the semester had the most significant impact on the students' final grades. Subsequently, ablation experiments were conducted, categorically combining the dataset into seven subsets, and using these subsets to train the six machine learning models separately. The results showed that using the feature category of student performance at various stages of the semester for predicting the

final grades was the most accurate. Simultaneously, the experiment concluded that using a dataset that integrates all three types of features would achieve the best predictive effect, further validating the rationality and effectiveness of the feature selection and combination in this study. At the same time, this study proposes a dynamically optimized teaching strategy solution that is applicable and feasible for the participants in the educational process (teachers and students).

Finally, the authors acknowledge that the limitations of this study primarily lie in the insufficient generalizability of the method due to the limited scope of the collected data. Specifically, in terms of the data on students' performance at different stages, this study only collected data from the C Programming course in the Software Engineering major at the authors' institution, which lacks sufficient support for the generalizability of the method. Future work will focus on enhancing the generalizability of the method. To improve the generalizability of the approach, future work should aim to collect data more widely from various majors and courses to construct the method.

**Author Contributions:** Conceptualization, Z.L. and J.M.; methodology, Z.L. and Y.D.; software, Z.L.; validation, Z.L., J.M. and Y.D.; formal analysis, Z.L., B.Q. and Z.Z.; investigation, Z.L.; resources, Z.L.; data curation, Z.L.; writing—original draft preparation, Z.L. and J.M.; writing—review and editing, Z.L. and J.M.; visualization, Z.L.; supervision, C.F.; project administration, D.K.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** Harbin Normal University Graduate Student Innovation Program under Grant No. HSDSSCX2024-42.

**Data Availability Statement:** The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request. The data on educational indicators of each province in the dataset are sourced from the official website of the Ministry of Education of the People's Republic of China ([http://www.moe.gov.cn/jyb\\_sjzl/moe\\_560](http://www.moe.gov.cn/jyb_sjzl/moe_560), accessed on 3 October 2024), and the data are authentic and reliable.

**Conflicts of Interest:** Author Zhanbo Zhu was employed by the company No. 703 Research Institute of China State Shipbuilding Company Limited. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Shen, Y.; Yin, X.; Jiang, Y.; Kong, L.; Li, S.; Zeng, H. *Case Studies of Information Technology Application in Education: Utilising the Internet, Big Data, Artificial Intelligence, and Cloud in Challenging Times*; Springer: Singapore, 2023.
2. Zhao, L.; Ren, J.; Zhang, L.; Zhao, H. Quantitative analysis and prediction of academic performance of students using machine learning. *Sustainability* **2023**, *15*, 12531. [[CrossRef](#)]
3. Peña-Ayala, A. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Syst. Appl.* **2014**, *41*, 1432–1462. [[CrossRef](#)]
4. Perkash, A.; Shaheen, Q.; Saleem, R.; Rustam, F.; Villar, M.G.; Alvarado, E.S.; de la Torre Diez, I.; Ashraf, I. Feature optimization and machine learning for predicting students' academic performance in higher education institutions. *Educ. Inf. Technol.* **2024**. [[CrossRef](#)]
5. Wang, X.; Zhao, Y.; Li, C.; Ren, P. ProbSAP: A comprehensive and high-performance system for student academic performance prediction. *Pattern Recognit.* **2023**, *137*, 109309. [[CrossRef](#)]
6. Grayson, A.; Miller, H.; Clarke, D.D. Identifying barriers to help-seeking: A qualitative analysis of students' preparedness to seek help from tutors. *Br. J. Guid. Couns.* **1998**, *26*, 237–253. [[CrossRef](#)]
7. Mengash, H.A. Using data mining techniques to predict student performance to support decision making in university admission systems. *IEEE Access* **2020**, *8*, 55462–55470. [[CrossRef](#)]
8. Baruah, A.J.; Baruah, S. Data augmentation and deep neuro-fuzzy network for student performance prediction with MapReduce framework. *Int. J. Autom. Comput.* **2021**, *18*, 981–992. [[CrossRef](#)]
9. Feng, G.; Fan, M.; Chen, Y. Analysis and prediction of students' academic performance based on educational data mining. *IEEE Access* **2022**, *10*, 19558–19571. [[CrossRef](#)]
10. Liu, C.; Wang, H.; Yuan, Z. A method for predicting the academic performances of college students based on education system data. *Mathematics* **2022**, *10*, 3737. [[CrossRef](#)]
11. Yue, L.; Hu, P.; Chu, S.-C.; Pan, J.-S. Multi-objective gray wolf optimizer with cost-sensitive feature selection for predicting students' academic performance in college English. *Mathematics* **2023**, *11*, 3396. [[CrossRef](#)]

12. Injadat, M.; Moubayed, A.; Nassif, A.B.; Shami, A. Multi-split optimized bagging ensemble model selection for multi-class educational data mining. *Appl. Intell.* **2020**, *50*, 4506–4528. [[CrossRef](#)]
13. Bansal, V.; Buckchash, H.; Raman, B. Computational intelligence enabled student performance estimation in the age of COVID-19. *SN Comput. Sci.* **2022**, *3*, 41. [[CrossRef](#)] [[PubMed](#)]
14. Asselman, A.; Khaldi, M.; Aammou, S. Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interact. Learn. Environ.* **2023**, *31*, 3360–3379. [[CrossRef](#)]
15. Pallathadka, H.; Wenda, A.; Ramirez-Asís, E.; Asís-López, M.; Flores-Albornoz, J.; Phasinam, K. Classification and prediction of student performance data using various machine learning algorithms. *Mater. Today Proc.* **2023**, *80*, 3782–3785. [[CrossRef](#)]
16. Zhang, T.; Liu, H.; Tao, J.; Wang, Y.; Yu, M.; Chen, H.; Yu, G. Enhancing Dropout Prediction in Distributed Educational Data Using Learning Pattern Awareness: A Federated Learning Approach. *Mathematics* **2023**, *11*, 4977. [[CrossRef](#)]
17. Van de Werfhorst, H.G.; Mijs, J.J.B. Achievement inequality and the institutional structure of educational systems: A comparative perspective. *Annu. Rev. Sociol.* **2010**, *36*, 407–428. [[CrossRef](#)]
18. Li, X. Education in China—Unbalanced Educational Development Caused by Regional Differences: Taking Gansu and Beijing (2010–2015) from Economic Perspective as Examples. *J. Educ. Humanit. Soc. Sci.* **2023**, *8*, 1441–1448. [[CrossRef](#)]
19. Kim, A.S.N.; Stevenson, C.R.; Park, L. Homework, in-class assignments, and midterm exams: Investigating the predictive utility of formative and summative assessments for academic success. *Open Scholarsh. Teach. Learn.* **2022**, *2*, 92–102. [[CrossRef](#)]
20. Ünal, F. Data mining for student performance prediction in education. In *Data Mining: Methods, Applications and Systems*; Birant, D., Ed.; IntechOpen: London, UK, 2020; pp. 423–432.
21. Jain, S.; Shukla, S.; Wadhvani, R. Dynamic selection of normalization techniques using data complexity measures. *Expert Syst. Appl.* **2018**, *106*, 252–262. [[CrossRef](#)]
22. Rani, P.; Vashishtha, J. An appraise of KNN to the perfection. *Int. J. Comput. Appl.* **2017**, *170*, 13–17. [[CrossRef](#)]
23. Nitesh, V.C. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321.
24. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
25. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T. LightGBM: A highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; pp. 3149–3157.
26. Rigatti, S.J. Random forest. *J. Insur. Med.* **2017**, *47*, 31–39. [[CrossRef](#)] [[PubMed](#)]
27. Hastie, T.; Rosset, S.; Zhu, J.; Zou, H. Multi-class adaboost. *Stat. Its Interface* **2009**, *2*, 349–360. [[CrossRef](#)]
28. Suthaharan, S. Decision tree learning. In *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*; Springer: Boston, MA, USA, 2016; pp. 237–269.
29. Jakkula, V. Tutorial on Support Vector Machine (SVM). *Sch. EECS Wash. State Univ.* **2006**, *37*, 3.
30. Assegie, T.A. Evaluation of the Shapley additive explanation technique for ensemble learning methods. *Proc. Eng. Technol. Innov.* **2022**, *21*, 20–26. [[CrossRef](#)]
31. Sahlaoui, H.; Alaoui, E.A.A.; Nayyar, A.; Agoujil, S.; Jaber, M.M. Predicting and interpreting student performance using ensemble models and shapley additive explanations. *IEEE Access* **2021**, *9*, 152688–152703. [[CrossRef](#)]
32. Yacoubi, R.; Axman, D. Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Online, 20 November 2020; pp. 79–91.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.