IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2, 2025

Predicative analysis of student performance in online learning

S.V. Durga Prasad[1], R. Lahari[2], P.Naga Sai[3], M. Pavana Sai[4], P. Harshavardhan[5]

[1] Assistant Professor, Department of Data Science Engineering, Chalapathi Institute of Engineering and Technology, Chalapathi Rd, Nagar, Lam, Guntur, Andhra Pradesh- 522034

[2,3,4,5] Students, Department of Data Science Engineering, Chalapathi Institute of Engineering and Technology, Chalapathi Rd, Nagar, Lam, Guntur, Andhra Pradesh- 522034

Email id: prasadsvd999@gmail.com[1], lahari.ravela@gmail.com[2], Thanmai thanmaipati2@gmail.com[3], pavansaimuttavarapu@gmail.com[4], pathuriharshavardhan@gmail.com[5]

Abstract

Student engagement plays a vital role in ensuring effective learning outcomes, especially in digital education settings. This study presents a machine learning approach to classify engagement levels as "engaged" or "not engaged" using the *Student Engagement Level-Binary* dataset. The Random Forest algorithm was utilized, achieving an exceptional accuracy of 100%, showcasing its ability to identify patterns and deliver highly reliable predictions. The trained model has been stored as “student engagement model.pkl”, enabling seamless deployment in real-time educational applications, such as engagement monitoring and adaptive learning platforms. These findings demonstrate the value of leveraging data-driven techniques to support personalized and timely interventions for improved learning experiences. Future research will focus on validating the model with larger datasets and exploring its integration into scalable, real-world systems.

Keywords: student engagement, binary classification, Random Forest, machine learning, educational technology, digital education, adaptive learning

1.Introduction

Student performance prediction research can be categorized into regression and classification problems, where the former predicts a score value, and the latter predicts a category [1]. Earlier studies in student performance prediction focused on traditional machine learning methods, which used logistic regression [2], support vector machines [3], and decision trees [4] to establish models for predicting student performance. In a study [5], an automatic method for observing and predicting student grades was proposed. This method utilized a genetic algorithm to capture the 30 best attributes from students’ historical learning data and trained a K-NN regression model and a decision tree using these features and labels to predict students’ performance score and categories. PEK et al. [6] used naive Bayes, random forest, decision tree, AdaBoost classifier, logistic regression, and KNN algorithms as basic learners, support vector machine (SVM) as meta-learner, and created a stacking method to develop a hybrid ensemble model. By analyzing the data, they discovered important features that influence student learning outcomes and successfully helped teachers identify students at risk. Jawad et al. [7] and Bujang et al. [8] combined the SMOTE technique with machine learning techniques to improve the impact of data imbalance on the model and enhance the accuracy of student performance prediction. Hung et al. [9] proposed a method for predicting student performance based on time-series clustering. The method aggregates learning behavior data such as the frequency of accessing course materials, frequency of reading forums, number of discussions,

and number of replies posted to identify at-risk students and predicts student performance more accurately than traditional frequency aggregation methods. Traditional machine learning methods ignore time information in the original data, which cannot effectively capture the impact of time features on student performance. However, deep neural networks can effectively address the problem of time information missing [10]. Therefore, some researchers have begun to utilize deep learning techniques to predict student performance. For example, He et al. [11] used a GRU network to extract time-series features from clickstream data and assessment score data and combined them with demographic features to predict student performance. Liu et al. [12] proposed a hybrid deep learning model that can extract time-behavioral and overall behavioral information from learning behavior data to more accurately predict high-risk students. Qu et al. [13] constructed a student performance prediction framework with an attention mechanism, in which an LSTM neural network was used to reflect students' learning processes, and a DSP-based adapter was used to enhance the importance of key information and improve the accuracy of student performance prediction. Kusumawardani et al. [14] proposed a transformer-based method for predicting student performance by converting the learning behavior data of students into a sequential feature vector. In some studies [15], authors used convolutional neural networks (CNN) to extract high-dimensional time information from the time series of student activities to better extract spatio-temporal features and utilized LSTM to capture the sequence information of student learning dynamics to more accurately predict student performance.

3.Methodology

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

XG boost

XG Boost 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. As a gradient boosting decision tree algorithment, the XG Boost algorithm model can be regarded as a decision tree additive strategy model, that is

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

Light GBM

Light GBM is similar to XG Boost 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 at the same time, Light GBM 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

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

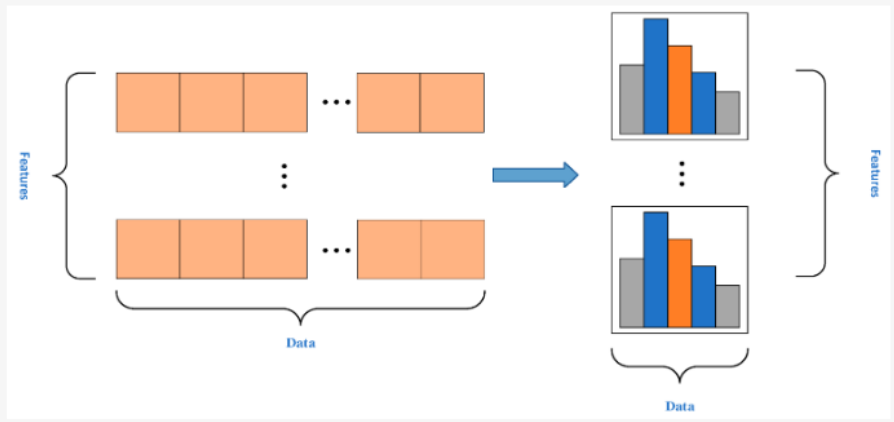


Figure: Schematic diagram of the Light GBM algorithm

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, Light GBM 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.

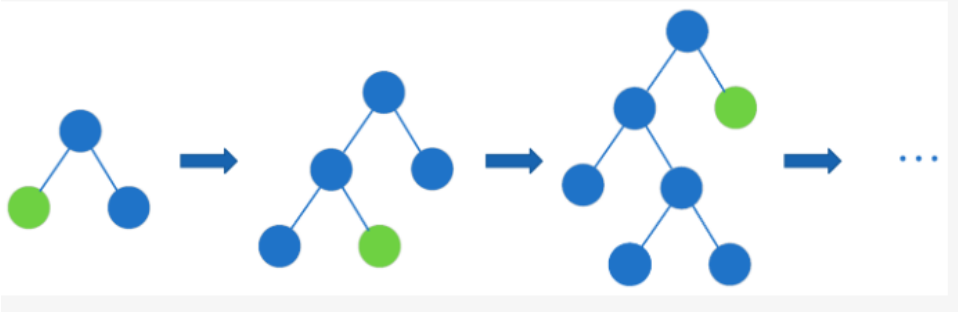


Figure: Schematic diagram of the Leaf-Wise algorithm

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.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

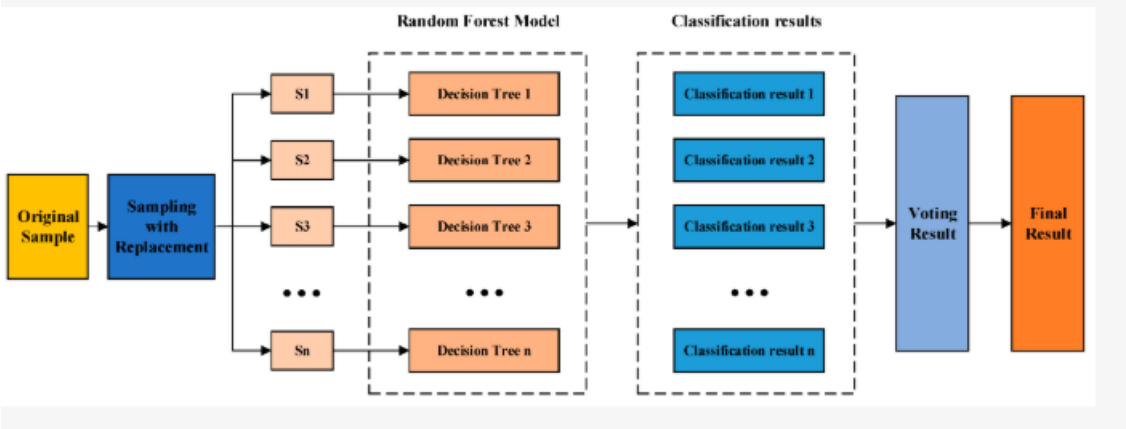


Figure: Random forest algorithm flow

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.

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.

Experimental 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 XG Boost, Light GBM, 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 The hyperparameter settings of the experimental models are described in Table 2, and the testing equipment and software are described.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

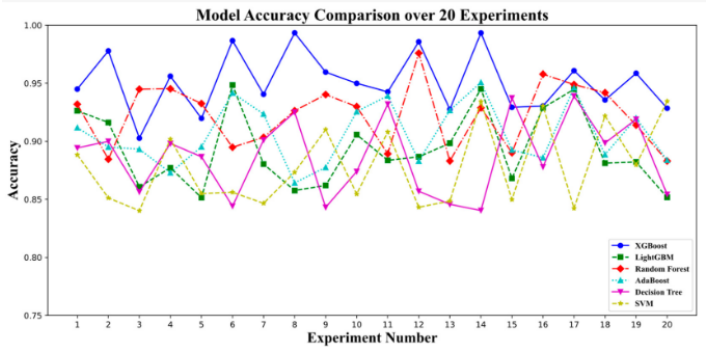
Table: Testing equipment and software

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

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:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$




```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
# Make predictions on the test data
y_pred = model.predict(X_test)
```

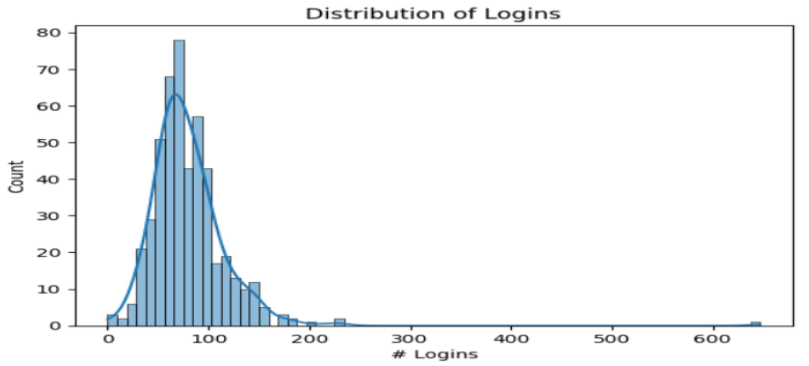
```
# Calculate and display accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

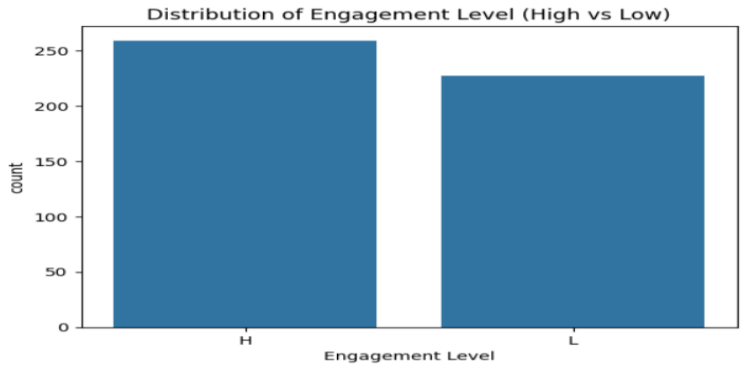
```
Accuracy: 100.00%
```

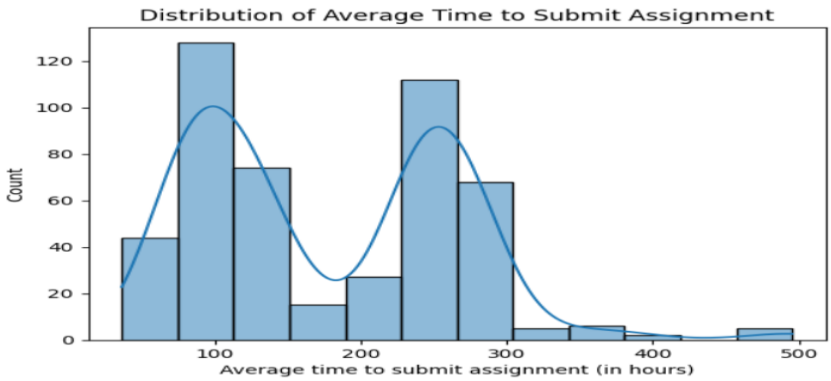
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

Figure: The accuracy of each model on the dataset with three types of features

	# Logins	# Content Reads	# Forum Reads	# Forum Posts	# Quiz Reviews before submission	Assignment 1 lateness indicator	Assignment 2 lateness indicator	Assignment 3 lateness indicator	Assignment 1 duration to submit (in hours)	Assignment 2 duration to submit (in hours)	Assignment 3 duration to submit (in hours)	Average time to submit assignment (in hours)
count	486.000000	486.000000	486.000000	486.000000	486.000000	486.000000	486.000000	486.000000	486.000000	486.000000	486.000000	486.000000
mean	79.897119	271.843621	2.156379	0.146091	2.045267	0.024691	0.024691	0.014403	227.659499	136.916324	168.520953	177.698925
std	41.293639	106.180726	8.898293	0.606881	1.964113	0.155343	0.155343	0.119269	96.342083	82.754479	101.934682	88.394268
min	0.000000	34.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	50.883333	6.200000	18.716667	36.327778
25%	58.000000	196.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	147.066667	58.708333	85.558333	99.620833
50%	74.000000	252.500000	0.000000	0.000000	2.000000	0.000000	0.000000	0.000000	191.033333	102.791667	128.133333	144.741667
75%	95.000000	338.750000	0.000000	0.000000	3.000000	0.000000	0.000000	0.000000	306.045833	212.112500	236.616667	250.500000
max	647.000000	1007.000000	58.000000	6.000000	12.000000	1.000000	1.000000	1.000000	558.000000	296.250000	632.000000	495.333333







Conclusions

This paper proposes a student performance prediction model (MTAPSP) based on multidimensional time-series data analysis. The model incorporates multidimensional data such as students' learning behaviors, assessment scores, and demographic information (e.g., age group, place of residence) to achieve the multi-classification prediction of students' performance, and the multi-classification prediction performance of the MTAPSP model is

verified by comparison with the baseline model. In addition, this paper tests the model's binary early prediction ability by dividing the dataset by course duration, and the experimental results show that it exhibits excellent performance in multi-classification prediction and binary early prediction. In future research, there is a need to address the imbalance in the data sample and to explore the impact of student–teacher interaction and student-to-student interaction data on student academic performance.

References:

1. Ahmad, M.S.; Asad, A.H.; Mohammed, A. A Machine Learning Based Approach for Student Performance Evaluation in Educational Data Mining. In Proceedings of the 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 26–27 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 187–192.
2. Zhang, W.; Huang, X.; Wang, S.; Shu, J.; Liu, H.; Chen, H. Student performance prediction via online learning behavior analytics. In Proceedings of the 2017 International Symposium on Educational Technology (ISET), Hong Kong, China, 27–29 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 153–157.
3. Al-Shehri, H.; Al-Qarni, A.; Al-Saati, L.; Batoaq, A.; Badukhen, H.; Alrashed, S.; Alhiyafi, J.; Olatunji, S.O. Student performance prediction using support vector machine and k-nearest neighbor. In Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 30 April–3 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–4.
4. Wang, C.; Wei, X.; Yang, A.; Zhang, H. Construction and Analysis of Discrete System Dynamic Modeling of Physical Education Teaching Mode Based on Decision Tree Algorithm. *Comput. Intell. Neurosci.* 2022, *2022*, 2745146.
5. Rao, G.M.; Kumar, P.K.K. Students Performance Prediction in Online Courses Using Machine Learning Algorithms. *United Int. J. Res. Technol* 2021, *2*, 74–79.
6. Pek, R.Z.; Özyer, S.T.; Elhage, T.; Özyer, T.; Alhajj, R. The role of machine learning in identifying students at-risk and minimizing failure. *IEEE Access* 2022, *11*, 1224–1243.
7. Jawad, K.; Shah, M.A.; Tahir, M. Students' Academic Performance and Engagement Prediction in a Virtual Learning Environment Using Random Forest with Data Balancing. *Sustainability* 2022, *14*, 14795.
8. Bujang, S.D.A.; Selamat, A.; Ibrahim, R.; Krejcar, O.; Herrera-Viedma, E.; Fujita, H.; Ghani, N.A.M. Multiclass prediction model for student grade prediction using machine learning. *IEEE Access* 2021, *9*, 95608–95621.
9. Hung, J.L.; Wang, M.C.; Wang, S.; Abdelrasoul, M.; Li, Y.; He, W. Identifying at-risk students for early interventions—A time-series clustering approach. *IEEE Trans. Emerg. Top. Comput.* 2015, *5*, 45–55.
10. Yu, C.C.; Wu, Y. Early warning system for online stem learning—A slimmer approach using recurrent neural networks. *Sustainability* 2021, *13*, 12461.
11. He, Y.; Chen, R.; Li, X.; Hao, C.; Liu, S.; Zhang, G.; Jiang, B. Online at-risk student identification using RNN-GRU joint neural networks. *Information* 2020, *11*, 474.
12. Liu, T.; Wang, C.; Chang, L.; Gu, T. Predicting High-Risk Students Using Learning Behavior. *Mathematics* 2022, *10*, 2483.

ISSN 2347–3657
Volume 13, Issue 2, 2025

13. Qu, S.; Li, K.; Wu, B.; Zhang, S.; Wang, Y. Predicting student achievement based on temporal learning behavior in MOOCs. *Appl. Sci.* 2019, *9*, 5539.
14. Kusumawardani, S.S.; Alfarozi, S.A.I. Transformer Encoder Model for Sequential Prediction of Student Performance Based on Their Log Activities. *IEEE Access* 2023, *11*, 18960–18971.
15. Chen, H.-C.; Prasetyo, E.; Tseng, S.-S.; Putra, K.T.; Prayitno; Kusumawardani, S.S.; Weng, C.-E. Week-Wise Student Performance Early Prediction in Virtual Learning Environment Using a Deep Explainable Artificial Intelligence. *Appl. Sci.* 2022, *12*, 1885.