Journal of Information Hiding and Multimedia Signal Processing ©2026 ISSN 2073-4212
Ubiquitous International Volume **17**, Number **1**, March **2026**

Analysis and Prediction of MOOC Learners' Learning Behavior Based on Machine Learning

Guoqing Yue[1,2,*]

[1]College of Computing and Information Technologies,
National University, Manila 1008, Philippines
[2]Anhui Sanlian University, Hefei 230601, P. R. China
80369657@qq.com

Mideth Abisado[1]

[1]College of Computing and Information Technologies,
National University, Manila 1008, Philippines
mbabisado@national-u.edu.ph

*Corresponding author: Guoqing Yue

Received July 30, 2025, revised September 17, 2025, accepted September 21, 2025.

Abstract. *With the rapid development of online education, massive open online courses (MOOCs) have become an important form of modern learning. However, how to effectively understand, analyze, and predict learners' learning behavior in the MOOC environment have always been a core concern in the field of educational research. This study utilizes the MOOCCube dataset to apply machine learning techniques to conduct an in-depth analysis of students' behavioral patterns and attempt to predict their future behavior. The research aims to provide educators and platforms with targeted information to better understand student behavior dynamics and develop effective teaching strategies, as well as optimize MOOC platform design and functionality. This research not only helps improve the effectiveness and satisfaction of MOOC education, but also has important value in improving the overall quality of online education.*
Keywords: MOOC; machine learning; learning behavior analysis; MOOCCube dataset; Online education.

1. **Introduction.** With the Internet and technology continuously advancing, massive open online courses (MOOCs) have become a global hot topic that profoundly impacts traditional education models. MOOCs provide high-quality academic resources to the world, making quality education accessible to everyone by lowering barriers to entry. However, this new form of education also presents challenges. Attracting and retaining students is an important issue for MOOCs while understanding and analyzing learners' behavior in the MOOC environment requires further research and exploration to improve teaching effectiveness. This study will conduct an in-depth analysis of the development process and challenges of MOOCs while exploring ways to enhance student participation and teaching effectiveness within these environments.

The purpose of this study is to utilize machine learning methods to comprehend and forecast the behavior of MOOC learners. The MOOCCube dataset, which contains a vast amount of MOOC learner behavior data, was used for in-depth analysis of students' learning behavior patterns [1]. By constructing and training machine learning models, we attempted to predict their future behavior. This approach aims to provide educators and platforms with targeted information that can help them better understand student learning dynamics, develop more effective teaching strategies, and optimize the design and functionality of MOOC platforms [2]. Our research not only strives to improve the educational effectiveness and satisfaction of MOOCs but also seeks to enhance the overall quality of online education.

To address these challenges effectively, we employed machine learning techniques for analyzing large-scale MOOC learner behavior data [3]. Our research objectives include analyzing such data based on

machine learning technology while exploring learners' behavioral patterns as well as identifying potential applications for machine learning technology in analyzing MOOC learner behaviors. We aim to put forward corresponding suggestions and improvement proposals.

With continuous advancements in science and technology, particularly within artificial intelligence (AI), various fields have widely adopted its branch known as "machine learning." Machine Learning has been especially useful in processing large-scale data sets while predicting future events across different domains including education where it predicts student achievement levels or recommends personalized paths towards achieving academic goals by analyzing their past performance records or current behaviors respectively.

These studies demonstrate that Machine Learning methods are effective tools that can assist educators in understanding learners' behavioral patterns while predicting their future actions thereby enabling them to develop personalized teaching strategies accordingly. Therefore, it is evident that Machine Learning's importance continues being explored within education with its potential role constantly expanding through practical application scenarios across diverse contexts [5].

2. **Literature review.**

2.1. **Research Status.** In the past period, many researchers have conducted in-depth research on the behavior patterns of MOOC learners. These studies revolve around user engagement, satisfaction, and learning outcomes. However, much of the research relies on traditional statistical methods to analyze learner behavior, which can pose significant challenges when dealing with large amounts of learner behavior data. In addition, studies that predict learners' future behavior are relatively rare.

Recent studies in the realm of Massive Open Online Courses (MOOCs) have illuminated distinct behavioral patterns among learners with varying levels of achievement [4]. Notably, individuals who demonstrate higher academic performance, including course completers and certificate achievers, are characterized by a heightened commitment to course activities and a more diversified approach to learning. These learners exhibit a structured engagement with course materials, actively participating in forums with content-focused discussions. Contrastingly, learners with lower performance often concentrate predominantly on video lectures, with a less varied learning process. Furthermore, research indicates that the learning sequences and interactive patterns of high-achieving learners display greater complexity and adherence to course schedules, as opposed to their lower-achieving counterparts. This body of work underscores the critical relationship between learner engagement, activity diversity, and academic achievement in MOOCs.

Kizilcec, R. F., Piech, C., & Schneider, E. (2013). Analyzing learner behavior in MOOCs led to the discovery of several distinct patterns of learner engagement. They used traditional statistical methods to classify MOOC learners into four typical participation trajectories: completers, auditors, continuous participants, and dropouts.

Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., & Seaton, D. T. (2013). An in-depth analysis of the first MOOCs on the edX platform emphasized the importance of MOOCs for education and research and explored their applications and potential. They employed traditional statistical methods [12].

Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., & Koller, D. (2013). The peer evaluation process in MOOCs was studied by building a machine learning model. A peer rating-based scoring model was proposed that can estimate student performance and grader bias and reliability.

Xing, W., Chen, X., Stein, J., & Marcinkowski, M. (2016). Machine learning methods, particularly ensemble learning, were used to predict whether learners in MOOCs would drop out. An ensemble machine learning approach was employed to propose a temporal prediction model to accurately and reliably predict students struggling in MOOCs.

Kizilcec, R. F., & Halawa, S. (2015). The key factors influencing online learner retention and achievement were revealed. Traditional statistical methods were used to investigate the characteristics and causes of attrition, providing an early identification strategy for identifying learners prone to attrition in MOOCs and guidance for targeting interventions [13].

Balakrishnan, G., & Coetzee, D. (2013). The use of the Hidden Markov Model (HMM) to predict MOOC learner retention. HMM was able to accurately predict student retention rates, providing the basis for personalized recommendations for online education.

Xing, W., Chen, X., Stein, J., & Marcinkowski, M. (2016). Predicting MOOC dropouts using stacked generalization. They used learning analytics and educational data mining techniques to analyze low-level structured data from catechisms to automatically infer student behavior and guide educational decisions.

Crossley, S., Paquette, L., Dascalu, M., McNamara, D. S., & Baker, R. S. (2016). Combining clickstream data with natural language processing (NLP) tools to better understand MOOC completion. The study combined clickstream data and NLP methods to examine whether students' online activity and language in discussion forums predicted successful course completion [6].

The existing research literature provides a rich set of methods for analyzing and predicting the behavior of MOOC learners, which provides an important theoretical foundation and methodological guidance for this study. These studies have focused on the behavioral patterns and learning outcomes of MOOC learners. For example, the study by Kizilcec et al. mainly uses traditional statistical methods to analyze learners' behavior, but they do not address the application of machine learning. In contrast, the research by Piech and Xing et al. attempts to use machine learning methods to analyze and predict learner behavior, such as peer evaluation processes in MOOCs and predicting learner dropout behavior. However, their research focused on specific learner behaviors without considering the overall behavioral patterns of MOOC learners.

In view of this situation, this research plan adopts machine learning methods to comprehensively analyze and predict the behavior of MOOC learners. The goal is to improve teaching effectiveness and learner satisfaction, which in turn will improve the overall quality of online education. By introducing machine learning methods, it is possible to deeply understand the behavior patterns of MOOC learners, predict their future behavior, and develop more effective teaching strategies accordingly [7].

2.2. **Common machine learning methods.** With the boom of big data and artificial intelligence, machine learning has been widely used in various fields. In particular, when it comes to processing large amounts of data and predicting future events, machine learning shows significant advantages. Education is no exception, with machine learning being used to predict student performance, recommend learning paths, and analyze student behavior [8]. These studies show that machine learning methods can effectively help educators understand students' behavior patterns, predict their future behavior, and develop personalized teaching strategies accordingly.

Table 1. Common machine learning methods for online learner behavior analysis

Machine learning methods	**Algorithm examples**	**Application in learning behavior analysis**
Classification method	Support vector machines (SVMs), decision trees, random forests, logistic regression	Predict whether learners will complete the course and drop out at some point in time
Clustering method	K-means, hierarchical clustering, DBSCAN	Discover learners' behavior patterns and group learners according to their behavioral characteristics
Sequence analysis	Hidden Markov Model (HMM), Sequence Pattern Mining, Recurrent Neural Network (RNN)	Predict learners' future behavior and understand how learners' behavior evolves
regression analysis	Linear regression, ridge regression, LASSO regression	Predict learners' academic performance, learning duration, etc
Neural networks	Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM)	Handle complex non-linear patterns to predict learner behavior and achievement

The above are some common machine learning methods and their application in learning behavior analysis, and the choice of which machine learning method to use depends on the research question and data characteristics. Machine learning has made some progress in education, but its application in the field of massive open online courses (MOOCs) is still relatively limited [9]. Therefore, this study attempts to use machine learning methods to deeply analyze and predict the behavior of MOOC learners, which is expected to provide strong theoretical support and practical guidance for research in related fields.

3. **Data Preprocessing.**

3.1. **Data Set.** The data used in this paper come from the MOOCCube dataset [1], an open source, large-scale data warehouse for MOOC-related research. Compared to other similar databases of educational resources, MOOCCube is large and contains a rich and diverse record of learning behavior. The dataset covers nearly 200,000 students' study time, study hours, and video viewing. This detailed student behavior data provides researchers with a rich and reliable resource for predicting course completion rates [10].

The MOOCCube dataset contains a large number of MOOC courses, videos, concepts, and learner behaviors of various types, such as 706 courses, 38,181 videos, 114,563 concepts, and 199,199 real users. In addition, a large-scale concept map and associated data are available. The entire MOOCCube data is clearly structured, and each part is thoroughly described and annotated for researchers to explore in depth. Among them, the course and student behavior records are all from the real environment of the "Academy Online" platform; the paper information comes from the academic search engine Aminer, and after automatic screening, crowdsourcing annotation, expert review and other processing stages, finally formed MOOCCube data warehouse.

TABLE 2. The entity types included in the MOOCCube dataset.

Entity Type	**Prefix**	**Important Fields**	**File**
Concept	K_	Name, English name, Explanation	concept.json
Course	C_	Name, About, Core ID, Video Order, Video Name, Chapter	course.json
Paper	P_	Title, Author, Venue, Abstract, Year, Num Citation	paper.json
School	S_	Name, About	school.json
Teacher	T_	Name, About	teacher.json
User	U_	Name, Course Order, Enroll Time	user.json
Video	V_	Name, Duration, Start and End Time, Text	video.json
Taxonomy	K_T_	Name	concept.json

3.2. **Preprocessing of MOOCCube datasets.** In this paper, the MOOCCube dataset is cleansed to remove duplicate data and irrelevant information to ensure data accuracy. With the next steps, we have a cleansed, transformed, and feature-engineered dataset that will help us more accurately analyze and predict learner behavior using machine learning models.

In terms data cleansing, this paper use Python's pandas library to read data and utilize the library's functions to delete incomplete records, such as records that are missing key fields. At the same time, the data is detected and processed for outliers, such as adjusting for watching videos for too long or too short.

TABLE 3. Preprocessing of MOOCCube datasets.

steps	**description**
1 Data cleansing	Ensure data accuracy by removing duplicate data and irrelevant information
2 Data conversion	For categorical variables, OneHotEncoder in Python's scikit-learn library is used to encode categorical variables into numeric variables that machine learning models can handle. For continuous variables, we normalize their distribution using StandardScaler to better align with the preset assumptions of the machine learning model.
3 Feature selection and construction	Based on previous research and domain knowledge, select some important features for subsequent learning behavior analysis. At the same time, some new characteristics are constructed based on existing features, such as learning activity (a combination of login frequency and video viewing time).

4. **Data analysis on machine learning.**

4.1. **Load data.** Use Python to implement the process of data loading, imports some commonly used data processing and machine learning libraries, including json, tqdm, defaultdict, pandas, matplotlib.pyplot, etc. In addition, models such as MultiLabel Binarizer, Support Vector Regression (SVR), Random Forest Regressor and GradientBoosting Regressor were imported from sklearn, and the dataset was divided into training and test sets using train_test_split functions [11].

4.2. **MOOC learner learning behavior exploratory analysis and visualization.**

4.2.1. *Popular video areas of MOOC..* Python's pandas and matplotlib libraries are used for data processing and visualization [14]. Use the Matplotlib library to plot a column chart showing the popularity of courses in different fields, where the x-axis represents the name of the course area and the y-axis represents the number of all courses in the field. The top three hottest areas are management science and technology, mathematics, and natural dialectics.

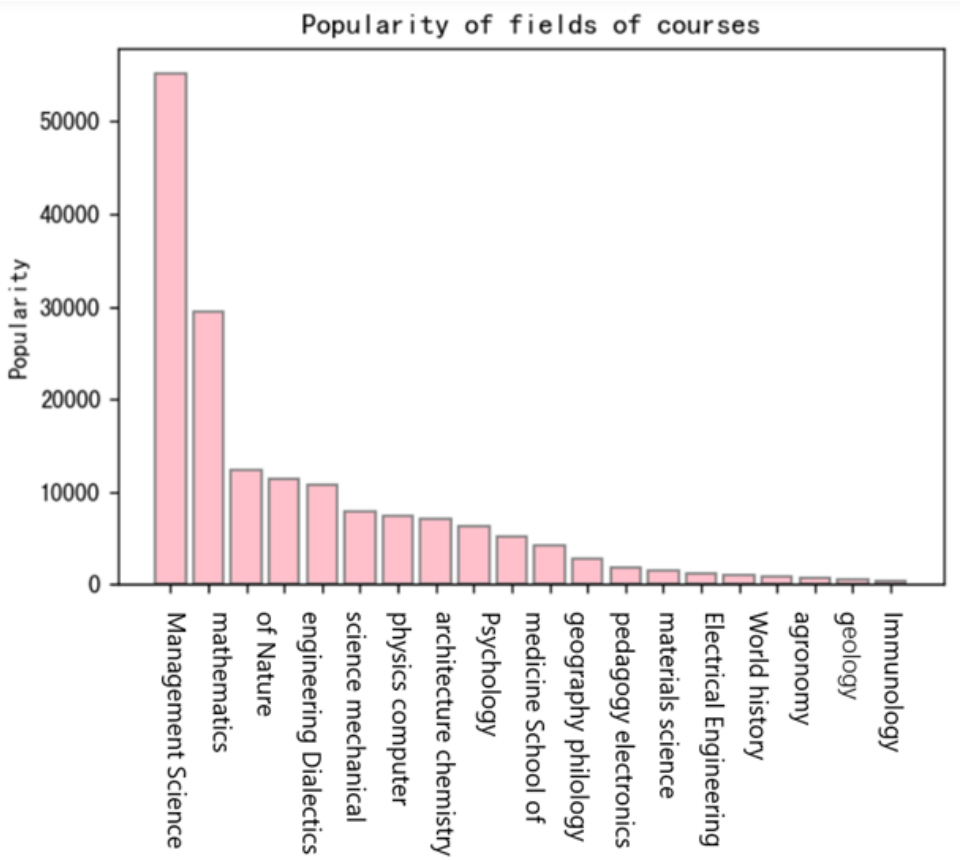

FIGURE 1. Popular MOOC videos .

4.2.2. *Word clouds of popular concepts.* The matplotlib library in Python was used to generate a word cloud. Word clouds show concepts such as abstraction, system, data, attributes, and so on are the most popular (Fig. 2).

4.2.3. *Video viewing time distribution.* Use Python's pandas and matplotlib libraries to plot the distribution of user viewing time over the course of the week. The horizontal axis represents the day of the week, and the vertical axis represents the number of views for that day. The statistics show that the distribution of when students log in and watch videos each week is relatively average, with most of the viewing occurring in the afternoon and evening hours of the day (Fig. 3, Fig. 4).

4.2.4. *Distribution of the number of courses viewed by students.* The matplotlib library in Python is used to plot histograms. Visualize the distribution of the number of courses students have watched (Fig. 5).

FIGURE 2. Word cloud of popular concepts

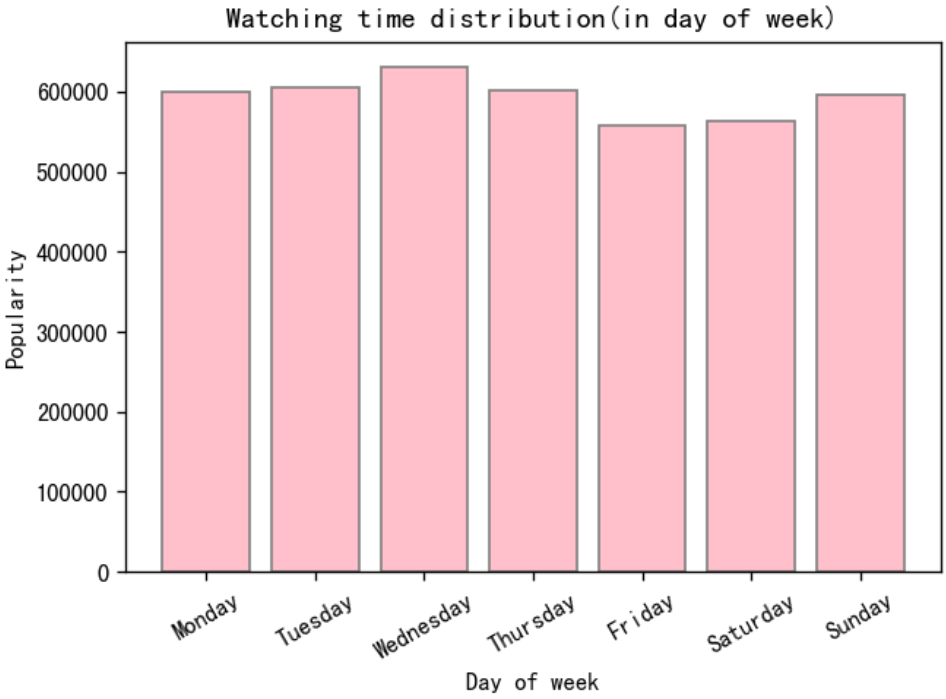

FIGURE 3. Distribution of video viewing time per week

4.2.5. *The average time interval between students joining a new course.* The pandas and matplotlib libraries are used to process user information and plot histograms [15]. First, filter out the information of users who have already taken at least two courses and process the registration time of these users. Sort each user's registration time from morning to evening. The number of days (i.e., time interval) spent between different courses for each user is then calculated and averaged as the average number of days spent across all courses for that user. Finally, a histogram is drawn with "Time Intervals" as the x-axis and "Frequency" as the y-axis, showing the distribution of the average number of days spent by all students before starting a new course. The statistics show that most of the students who have enrolled multiple times will enroll in new courses within tens of days of the previous enrollment.

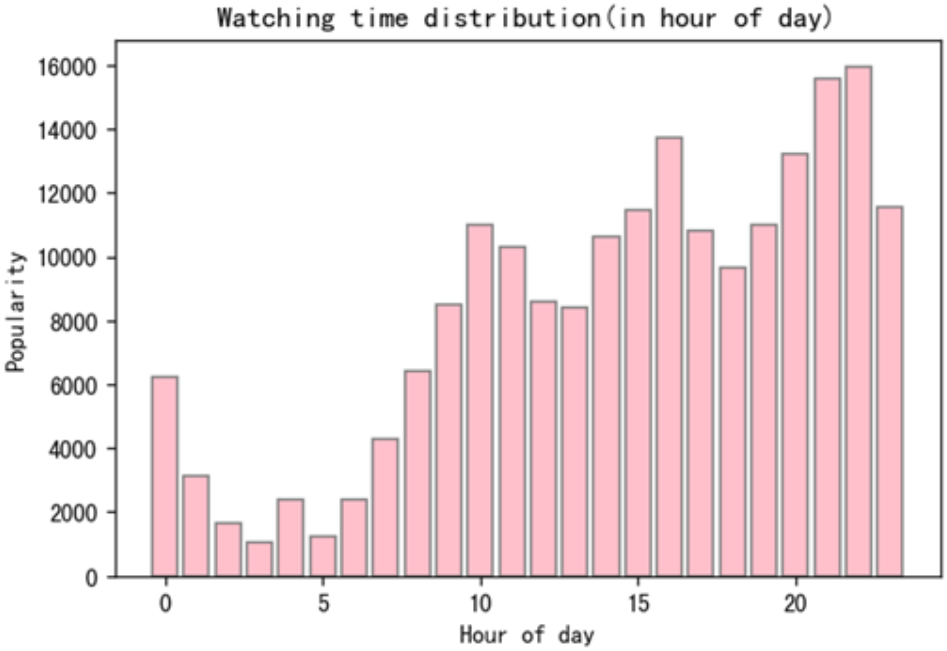

FIGURE 4. Distribution of video viewing time per day

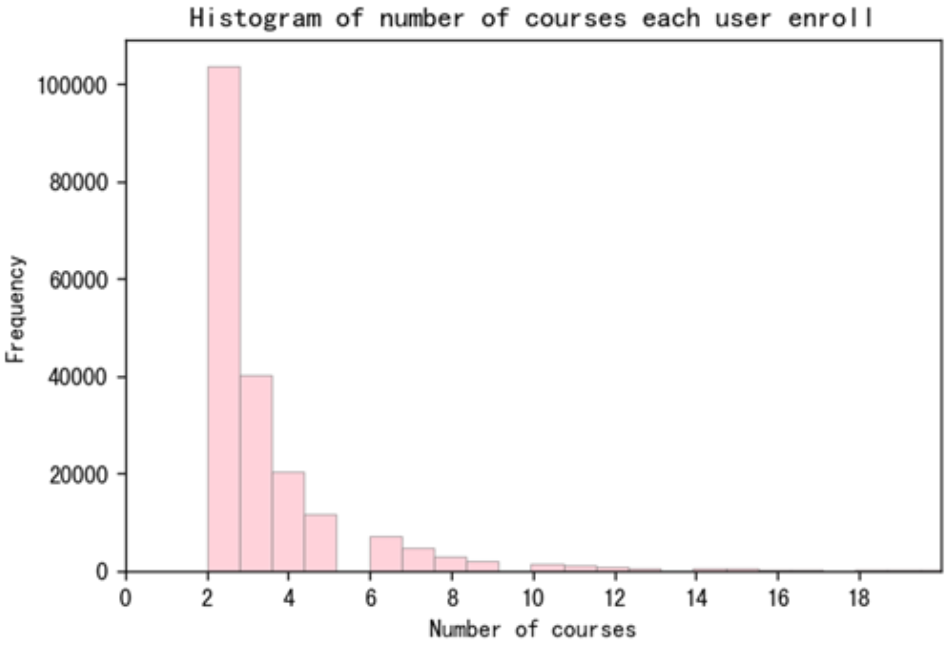

FIGURE 5. Distribution of video viewing time per day

4.2.6. *Average video completion rate distribution per student.* Using the Python language, it is used to calculate the average completion rate of videos watched by users and draw a histogram to show the distribution of average video completion rates for each user. First, the "video_progress_time" (watched time) in the user's viewing activity data is divided by the "video_duration" (total time) to obtain the completion rate of each user's corresponding video. Then, the lambda function is used to set all values greater than 1 to 1, i.e., the maximum value is 1. Next, each user ID is grouped and its average completion rate is calculated. Finally, the Matplotlib library is used to plot histograms to show how many users are in different average completion rate intervals. The statistics show that students generally have higher completion rates for course videos.

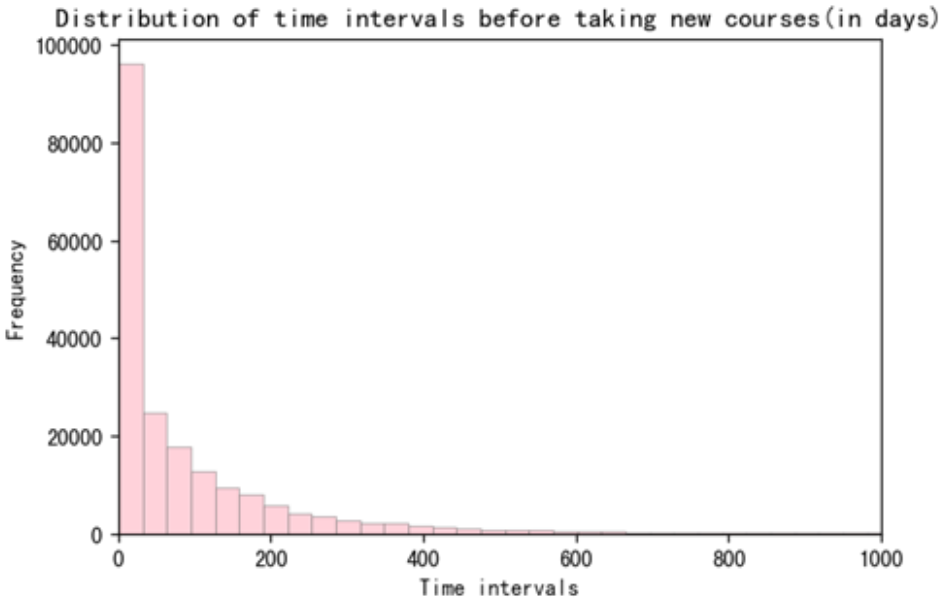

FIGURE 6. Average time interval for students to join a new course

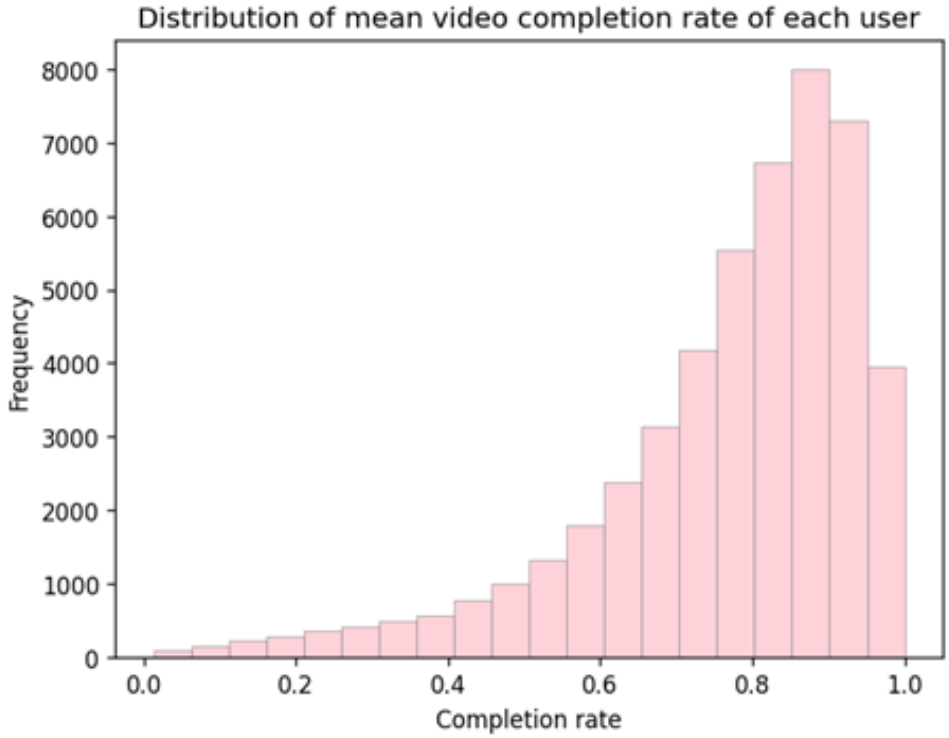

FIGURE 7. Average video completion rate distribution for each student

4.2.7. *Average distribution of views per video per student.* The matplotlib library in Python was used to plot the distribution histogram of video views [16]. First, it selects the 'id' and 'watching_count' columns from the data frame named user_video_act and calculates the average by grouping the 'id' to get the average mean_watching_count of each user's views. Then, it uses the plt.hist() function to plot the histogram and sets some parameters such as bins, color, edgecolor, etc. to beautify the plot. Finally, it adds x-axis labels, y-axis labels and titles using the plt.xlabel(), plt.ylabel() and plt.title() functions.

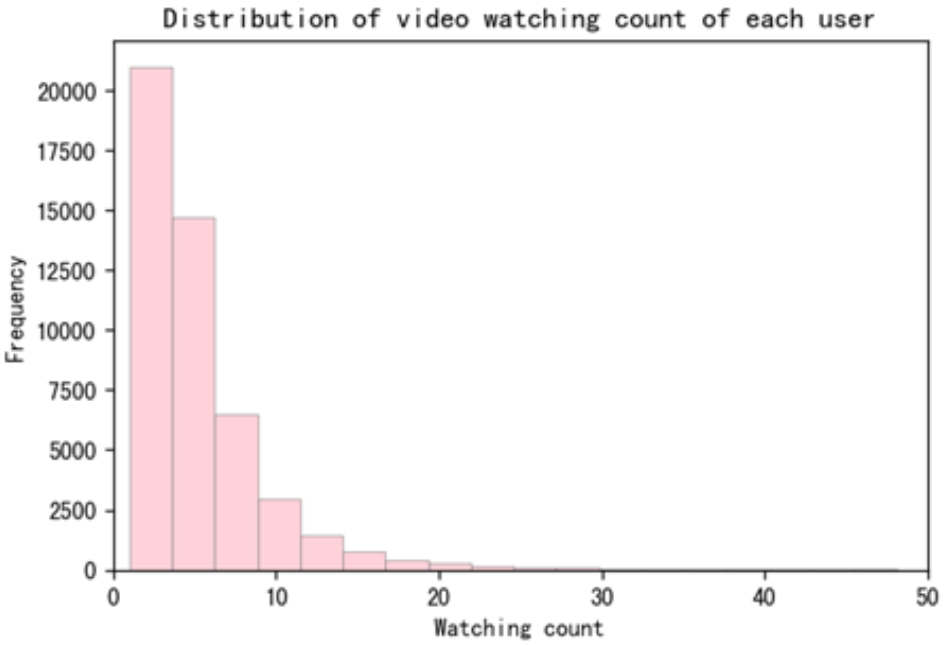

FIGURE 8. Average number of views per video per student

TABLE 4. Feature extraction.

Video dimension	The realm of video (multi-hot).
The realm of video (multi-hot).	The number of videos watched
The school where the course is located (one-hot).	The number of courses watched
The first few of the current courses	Average watch time for videos
Video length	The actual duration of the video
	The proportion of schools where the course was viewed
	The proportion of the various areas in which the course is viewed
	The average number of videos watched per course
	Average completion rate of watched videos

4.2.8. *Feature Extraction.* Table 4 describes the feature extraction process for video-based datasets. The following is an explanation of each feature:

Video dimension: This characteristic indicates the domain or category to which the video content belongs. It is encoded using multi-hot encoding, where each dimension corresponds to a specific video domain. If a video belongs to more than one domain, the corresponding dimension is activated (set to 1), indicating that these domains exist in the video.

Number of videos viewed: This characteristic represents the total number of videos watched by an individual. It represents the overall engagement of users with video content.

Course School: This feature is a one-hot coded representation of the school where the course is located. Each school is assigned a unique dimension, and the dimension corresponding to the school where a particular course is located is activated.

Number of courses watched: This characteristic represents the total number of courses watched by an individual. It reflects user engagement with different courses.

Top few of the current course: This feature captures information about the course the user is currently taking. It may be encoded using the appropriate method.

Average watch time of video: This characteristic represents the average length of time a user watches a video. It provides insights into the user's viewing habits and attention span.

Video Length: This characteristic represents the actual duration of the video. It represents the length of each video in the dataset.

Proportion of courses viewed in different schools: This characteristic represents the distribution or proportion of courses viewed in different schools to which the video is viewed. It helps to understand the diversity of schools where users consume content.

Proportion of courses viewed in different regions: This characteristic indicates the distribution or proportion of courses viewed in different regions or geographies to which the video is viewed. It provides insights into the geographical distribution of users' learning interests.

Average number of videos watched per course: This feature calculates the average number of videos watched per course. It indicates how engaged the user is with the content of each course.

Average completion rate of watched videos: This characteristic represents the average rate at which users complete the video they started watching. It reflects the user's level of engagement and commitment to completing the view.

These characteristics provide a variety of information about users' video consumption behavior, preferences, and level of participation in different courses and schools.

4.3. **Prediction of Video Completion Rate.**

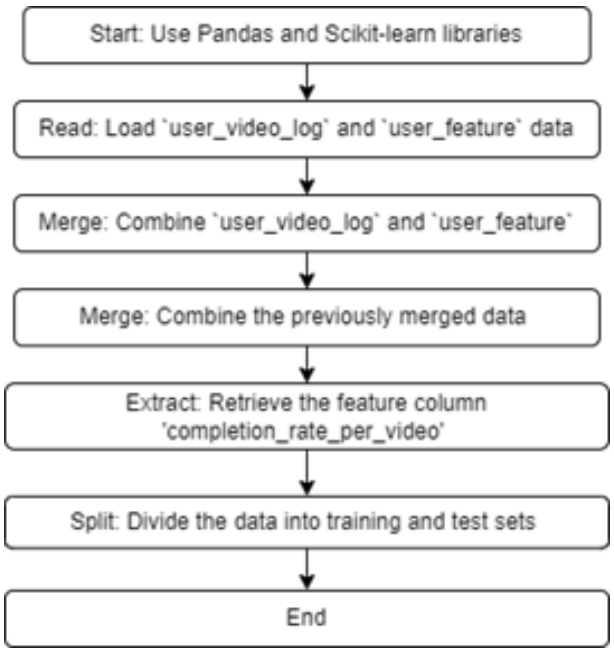

FIGURE 9. Dataset Division

4.3.1. *Data Split.* The Python code uses the Pandas library and the Scikit-learn library. First, it merges the two data frames, "user_video_log" and "user_feature", according to a common column "id", with the suffixes "_per_video" and "_per_user" respectively. It then merges another data frame "video_feature" with the previously merged data frame according to the common column "video_id", with the suffixes "_per_user" and "_per_video" respectively.

Next, extract a column of features named "completion_rate_per_video" from the newly generated "user_video_log" data frame as the target variable (y), and delete the column in which the feature is located. Finally, use the train_test_split function in Scikit-learn to divide the training and test sets into training and test sets as independent variables (X), and return four objects: training set arguments, test set arguments, training set target variables, and test set target variables. where test_size=0.2 means that the test set accounts for 20% of the total number of samples, and random_state=42 means that the random seed value is set to 42.

```
from sklearn.linear_model import LinearRegression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
y_pred = lr_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error (MSE) of LR:", mse)
```

```
Mean Squared Error (MSE) of LR: 0.07294034517362842
```

FIGURE 10. Linear regression model prediction

4.3.2. *Linear Regression Model Prediction.* The Python code uses the linear regression model from the "sklearn" library to make predictions. First, the "LinearRegression" class was imported and an instance object "lr_model" was created. The model is then fitted using the training datasets (X_train and y_train). Next, make predictions with the test data set X_test and save the results in variable y_pred. Finally, the Mean Squared Error (MSE) is 0.07294, which is the average quadratic deviation between the true value and the predicted value, is calculated and output.

```
rf_model = RandomForestRegressor()
rf_model.fit(X_train, y_train)
y_pred = rf_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error (MSE) of random forest:", mse)
```

```
Mean Squared Error (MSE) of random forest: 0.07664661178907446
```

FIGURE 11. Random forest model prediction

4.3.3. *Stochastic forest regression model predictio.* This code uses a random forest regression model to predict the outcome of the test set and calculates the mean squared error (MSE) between the predicted and true values. Among them, "RandomForestRegressor()" is a random forest regression model object, the "fit()" method is used to train the model and fit the data, and the "predict()" method is used to make predictions on the test set. Finally, the value of MSE is output. The value is 0.0766.

```
from sklearn.tree import DecisionTreeRegressor
dt_model = DecisionTreeRegressor()
dt_model.fit(X_train, y_train)
y_pred = dt_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error (MSE) of decision tree:", mse)
```

```
Mean Squared Error (MSE) of decision tree: 0.14387148458774773
```

FIGURE 12. Decision tree model prediction

4.3.4. *Decision Tree Regression Model Prediction.* This code uses the decision tree regression model from the "sklearn" library to make predictions. First, the "DecisionTreeRegressor" class was imported and

an instance object "dt_model" was created. Then, use the training datasets (X_train and y_train) to fit the model. Next, make predictions with the test data set X_test and save the results in variable y_pred. Finally, the mean squared error between the true and predicted values is calculated and the result is output. The value is 0.14387.

Among these three prediction methods, the linear regression model has the highest prediction accuracy with a mean squared error (MSE) value of 0.07294. In comparison, the MSE values for random forest regression and decision tree regression are 0.0766 and 0.14387 respectively, indicating slightly inferior performance. Based on the characteristics of this prediction task, the linear regression model is particularly suitable for it due to the following reasons:

The linear regression model is based on the assumption that there is a linear relationship between output results y and input features x, which is reasonable when predicting completion rates for video learning courses. However, non-linear assumptions in decision tree and random forest models may not be entirely applicable in such tasks.

The parameters of the linear regression model are relatively simple, consisting only of a set of slope and intercept values; therefore it is easy to train and adjust. Conversely, decision tree and random forest models have more parameters which can lead to overfitting resulting in relatively poor predictive performance.

The linear regression model is relatively stable with strong consistency in fitting results across different training sets. However, due to randomness effects, decision tree and random forest models may exhibit instability in their results.

Overall, for this video learning course completion rate prediction task, among these three methods, the effectivity of the Linear Regression Model is outstanding. However if more features could be combined into building models, such as viewing times, studies duration, wrong question numbers etc., the predictive effect might become even more accurate. For more complex prediction tasks, the Decision Tree or Random Forest Models may show better performance. However, a simple yet efficient Linear Regression Model remains an important method in machine learning, and applies to many predictive analysis scenarios.

5. **Conclusion.** This study aims to deeply analyze and predict the learning behavior of MOOC learners. In achieving this goal, the MOOCCube dataset was selected as the data source, and Python was used for a series of data preprocessing, including data cleaning, data transformation, and feature selection and construction. In the data analysis stage, descriptive statistical analysis, hypothesis testing, correlation analysis, and further visual analysis were carried out on learners' behavior. To further predict learner behavior, we selected machine learning models to predict learners' video completion rates.

Based on the results of our experiment, this paper finds that learners' login frequency and video viewing time have a significant positive correlation with their course completion rate. At the same time, learners' participation in the forum also showed a significant positive correlation with the course completion rate. These findings clearly reveal the strong connection between learners' learning behavior and learning outcomes, which is an important reference value for educators and educational platforms in understanding and guiding learners' learning behavior.

In terms of predicting learners' video completion rate, this paper compares the prediction effects of multiple machine learning models and finds that the random forest regression model has the best prediction effect. This finding further proves the effectiveness of machine learning methods in predicting learners' learning behavior, which is of great significance for optimizing the allocation of teaching resources and improving teaching effects.

Overall, this paper delves into the learning behaviors of MOOC learners and effectively predicts these behaviors by applying machine learning techniques. The research in this paper can not only improve the teaching effect and learner satisfaction of MOOC, but also contribute important theoretical and practical value to improving the overall quality of online education.

Future research in this field can expand in multiple potential directions, including:

Expanding the dataset: This study is based on the MOOCCube dataset, but incorporating more diverse datasets may help gain a more comprehensive perspective. Different MOOC platforms may have different user behavior patterns, and understanding these patterns could provide broader insights.

In-depth analysis of features: This study identified key features such as login frequency, video watching time, and forum participation that are strongly correlated with course completion rates. Future research can further deepen the analysis of these features or explore other possible ones such as learners' login times, their speed through courses, and their interactions with peers.

Application of advanced machine learning models: Although in this study linear regression model showed the best predictive performance, future research can explore using more advanced models including

deep learning methods. For example, considering the common existence of time-series data when studying user behavior, sequence models like LSTM (Long Short-Term Memory) or Transformer could be very beneficial.

Developing personalized learning paths: Based on insights derived from machine learning models, future research can focus on creating more personalized learning paths. For instance frequent logins learners might benefit from course structures that break content into smaller digestible parts while those who watch videos for longer periods might prefer deeper and comprehensive courses.

Explore the impact of demographic factors: Future work can explore how demographic factors such as age group, geographic location, and educational background affect learner behavior and course completion rates. This could provide further insight into how to better personalize and optimize online educational experiences.

By exploring these potential research directions listed above, we can further enrich our understanding of online learning behaviors and thereby improve the overall quality of online education.

Acknowledgment. This work funded by the Anhui Provincial Natural Science Project 2022AH051993.

REFERENCES

[1] Jifan Yu, et al. "MOOCCube: A large-scale data repository for NLP applications in MOOCs." *Proceedings of the 58th annual meeting of the association for computational linguistics.* 2020.

[2] Abisado, Mideth B., et al. "A flexible learning framework implementing asynchronous course delivery for Philippine local colleges and universities." *International Journal* 9.1.3 (2020).

[3] Kizilcec, René F., Chris Piech, and Emily Schneider. "Deconstructing disengagement: analyzing learner subpopulations in massive open online courses." *Proceedings of the third international conference on learning analytics and knowledge.* 2013.

[4] Li, Shuang, Junlei Du, and Jingqi Sun. "Unfolding the learning behaviour patterns of MOOC learners with different levels of achievement." *International Journal of Educational Technology in Higher Education* 19.1 (2022): 22.

[5] Radwan, Ghada Abdallah, et al. "Coronary Artery Disease Prediction by Combining Three Classifiers." *Journal of Information Hiding and Multimedia Signal Processing* 15.4 (2024): 221-235.

[6] Xing, Wanli, et al. "Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization." *Computers in human behavior* 58 (2016): 119-129.

[7] Kizilcec, René F., and Sherif Halawa. "Attrition and achievement gaps in online learning." *Proceedings of the second (2015) ACM conference on learning@scale.* 2015.

[8] Balakrishnan, Girish, and Derrick Coetzee. "Predicting student retention in massive open online courses using hidden markov models." *Electrical Engineering and Computer Sciences University of California at Berkeley* 53 (2013): 57-58.

[9] Crossley, Scott, et al. "Combining click-stream data with NLP tools to better understand MOOC completion." *Proceedings of the sixth international conference on learning analytics & knowledge.* 2016.

[10] Yuan, Li, and S. J. Powell. "MOOCs and open education: Implications for higher education." (2013).

[11] Breslow, Lori, et al. "Studying learning in the worldwide classroom research into edX's first MOOC." *Research & Practice in Assessment* 8 (2013): 13-25.

[12] Liang, Jiajun, Chao Li, and Li Zheng. "Machine learning application in MOOCs: Dropout prediction." *2016 11th International conference on computer science & education (ICCSE).* IEEE, 2016.

[13] Kloft, Marius, et al. "Predicting MOOC dropout over weeks using machine learning methods." *Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs.* 2014.

[14] Sial, Ali Hassan, Syed Yahya Shah Rashdi, and Abdul Hafeez Khan. "Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python." *International Journal* 10.1 (2021): 277-281.

[15] Hetland, Magnus Lie, and Fabio Nelli. "Activity 1: Data Analysis with Pandas, Matplotlib, and Seaborn." *Beginning Python: From Novice to Professional.* Berkeley, CA: Apress, 2024. 487-504.

[16] Dol, Sunita M., and P. M. Jawandhiya. "Data Visualization for the Dataset Collected from Education Sector Using Python." *2024 1st International Conference on Communications and Computer Science (InCCCS).* IEEE, 2024.