This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2881275, IEEE Access

IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

An Integrated Framework with Feature Selection for Dropout Prediction in Massive Open Online Courses

LIN QIU[1,2], YANSHEN LIU[3], AND YI LIU[3]
[1] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, Hubei 430079, China (e-mail: ql@mails.ccnu.edu.cn)
[2] School of Computer Science, Yangtze University, Jingzhou, Hubei 434023, China
[3] Educational Informatization Research Center of Hubei, Central China Normal University, Wuhan, Hubei 430079, China

Corresponding author: Lin Qiu (e-mail: ql@mails.ccnu.edu.cn).

This research is supported by Educational Informatization Research Center of Hubei, Central China Normal University.

ABSTRACT Massive Open Online Courses (MOOCs) have flourished in recent years, which is conducive to the redistribution of high-quality educational resources globally. However, the high dropout rate in the course of operation has seriously affected its development. Therefore, in order to improve the degree of completion, it is an effective way to study how to effectively predict the dropout in MOOCs and intervene in advance. Traditional methods rely on manually extracted features, which is difficult to guarantee the final prediction effect. In order to solve this problem, this paper proposes an integrated framework with feature selection (FSPred) to predict the dropout in MOOCs, which includes feature generation, feature selection, and dropout prediction. Specifically, FSPred applies a fine-grained feature generation method in days to generate features and then uses an ensemble feature selection method to select valid features and feed them into a logistic regression model for prediction. Extensive experiments on a public dataset have shown that FSPred can achieve the comparable results with other dropout prediction methods in terms of precision, recall, F1 score and AUC score. Finally, through the analysis of the features of the final selection, the suggestions for the construction of the MOOCs are put forward.

INDEX TERMS Dropout Prediction, Feature Generation, Feature Selection, MOOCs

I. INTRODUCTION

WITH the rapid development of Massive Open Online Courses (MOOCs) in the world, it provides more opportunities for people to learn, so that people can learn high-quality courses anytime and anywhere, to some extent make up for the uneven distribution of educational resources and improve the fairness of education. However, with the application of MOOCs, many problems have been exposed, such as the low degree of completion of courses, the query of learning effect and so on, [1]. How to solve these problems, so that the MOOCs can go further, and bring more educational benefits is a hot problem that the MOOC platform, MOOC teachers, and MOOC researchers pay close attention to [2]. Among them, improving the completion degree of the courses can improve the learning effect of the students to a certain extent. Therefore, the primary question is how to enable more students to continue online learning, so that they can spend more time to learn and achieve greater learning effect. One of the effective ways to solve this problem is to find the students who have the risk of dropping out in advance, that is, to predict the dropouts and conduct a certain degree of intervention.

The existing methods for predicting the dropout in MOOCs are based on data mining, that is, the feature extraction is performed on various data generated by students during the study, and then various mining algorithms are used to make the final predictions [3]– [11]. Students will produce various types of data records when learning MOOCs, including basic demographic information about students [7], learning clickstream data (video playback information, module access information, etc.) [11], forum posts and interactive information [9], test interaction and performance information [5] and so on. Different researchers used one or more types of data to predict whether students might drop out of MOOCs. And the experiments they have done shown that each of these information has some effect. The learning clickstream data representing the student's learning process behavior is the most widely used [3], [5]– [6], [8]– [11] because this kind of data has the most extensive coverage. Basically, the participants who participate in MOOCs learning will produce clickstream data more or less. And this kind of data reflects more details of learners' learning behavior and can be used to better predict whether learners will drop out in the future by mining their learning behavior patterns. After a large number of features have been manually extracted from these data, they will be fed into the corresponding classifier for classification. The extracted features are usually some coarse-grained statistical information, such as [3], [5]– [6], etc., mainly based on the weekly time unit as a whole for statistical calculation. The effects of the features extracted by these methods depend on the domain knowledge of the extractors. Generally, the number of features extracted is small, and the details of the data have not been fully utilized, then a certain loss of the prediction effect has been caused.

In order to solve these problems, this paper proposes an integrated prediction framework based on feature selection (FSPred). In this framework, we first perform a fine-grained (one day as a unit of time) feature generation method to achieve conversion of data details and to generate candidate feature sets. However, the number of features generated by a finer-grained manner will become larger. These features may contain some redundant features, which may increase the computational overhead, reduce the accuracy, and cause a certain bias in the prediction results. Therefore, we use an ensemble feature selection method to sort and select the features, to eliminate the redundant features, and to form the final subset of features used for prediction. Finally, the test set is mapped to the same feature set and the final prediction is given. The ensemble feature selection method is an important part of FSPred for the effectiveness and cost.

The main contributions of this paper are:

- We propose a statistical analysis of learning behavior in the unit of one day so that the granularity of feature generation is finer, and the loss of original information in the process of generation is reduced.
- The ensemble feature selection method is introduced to eliminate the problems of precision degradation and computational cost caused by redundant features, and detailed experiments on a real dataset demonstrate the effectiveness of our proposed method.
- A detailed analysis of the experimental results has been carried out, and the behaviors that mainly affected learners' dropouts have been given.

The remaining of this paper is organized as follows. In Section II we review the related work. Section III presents our proposed FSPred framework and the details of its algorithms. Section IV illustrates the experimental results on the real data set and our discovery of them. Section V summarizes the entire study.

II. RELATED WORKS

In this section, we review the existing methods for predicting MOOCs students' dropouts and the various feature selection methods that are commonly used.

A. DROPOUT PREDICTION IN MOOCS

1) Definition of dropout

The definitions of MOOC students' dropouts are mainly divided into two categories in the existing literature. One is whether the students finally complete the course and obtain the certificate as the judgment standard, if without completing the course and obtaining the certificate, the student is dropout [12]; the other is whether a student has a learning behavior for a period of time is a criterion for judging, that is, if he/she has not entered the MOOCs for a period of time then the student is to drop out of the courses [10]. In order to detect the dropout risk of students as early as possible, we adopt the second way to define dropouts. This is consistent with the definition of dropouts of the selected dataset used in the following experiments. We take students who drop out as positive examples, and students who do not drop out as negative examples.

2) Prediction methods

From the definition of the problem, the dropout prediction problem is a typical binary classification problem. Most of the existing studies have been conducted by standard supervised learning methods. However, in the specific implementation, the dropout prediction problem is divided into two types. One is to regard the problem as a classic binary classification problem, use support vector machine (SVM) and logistic regression (LR) to realize learning and prediction. In particular, logistic regression is easy to understand, simple to operate, and widely used. For example, in [6], a MOOC's clickstream data of different weeks was used to train an SVM classifier to predict the students' dropout. Literature [13] trained a logistic regression classifier to predict the probability of students completing the course and obtaining a certificate in combination with the task performance and social interaction of the first week of a MOOC course on Coursera. The work in [8] applied the clickstream data and the forum submission data of the course 6.002x on edX to train a logistic regression classifier to predict whether students would stop learning in the next week. Literature [14] adopted LR to analyze the course completion, assignment completion and scoring on Coursera to make early dropout prediction for early intervention. Another is to focus on the time series characteristics of the data and consider the prediction problem as a time series classification problem, which can be solved by hidden Markov chain, Nonlinear State Space Model, and RNN. For example, the work in [3] used a Hidden Markov Model and multiple types of data on edX, such as the clickstream data, the access and release of the forum, and the evaluation of the students, to mining the students' behavior patterns, and to predict whether students would drop out next week. Literature [15] applied a Nonlinear State Space Model to predict student dropouts by combining clickstream data from different weeks. The work in [10] used Recurrent Neural Network (RNN) and HMM to predict whether students would drop out with clickstream data from some weeks. Literature [16] proposed a temporal modeling method for student dropout behavior. The work in [17] applied a deep neural network model combining Convolutional Neural Networks and Recurrent Neural Networks to make dropout prediction. Literature [18] compared and analyzed the above two categories of methods and gave their respective advantages.

Some scholars have begun to pay attention to the impact of dataset on prediction results. The work in [19] studied the relationship between the dataset used for prediction and the prediction effect, and derived the characteristics of the proposed dataset. Literature [20] studied the data subset selection problem of multi-MOOC level dropout prediction problem, and proposed a subset selection model and verified it in neural network. These studies are mainly for the selection of training data and do not address specific feature processing issues. In the existing researches of dropout prediction in MOOCs, there are only dozens of features extracted from the raw data. The number of features is small and the quality of the extracted features is related to the professional domain knowledge and experience of the researchers. There may be a loss of the original data details. Therefore, finding a feature extraction method that is more general and simple without relying on professional domain knowledge to retain more detailed features can make the prediction of dropout problem in MOOCs more stable and accurate.

B. FEATURE SELECTION

When the feature dimension of the data to be classified is high, there may be some redundant features or noise features in the feature set. In this case, feature selection can be used to eliminate these features, so as to select the smaller optimal subset to make the learning model achieve better recognition performance and reduce computational complexity. Feature selection methods can be divided into four categories by means of feature subset evaluation strategies of existing researches.

(1) Filter method [21]. It is mainly based on the correlation between the features and the classes, and then all the features whose correlation is greater than the set threshold are selected to constitute the final feature subset. This method has low computational complexity and does not depend on a specific learning algorithm, so it has a wide range of adaptability, but it cannot achieve a stable and ideal performance of a specific algorithm. Correlation calculation methods used in this type of method mainly include: chi-square (Chi2) test [22], Pearson correlation coefficient [21], mutual information [23], and maximum information coefficient (MIC) [24], and distance correlation coefficient [21], etc.

(2) Wrapper method [25]. The prediction method is introduced to score the prediction effect of the feature subset on this method, and the feature subset with the optimal performance is selected as the final selection result. This method is related to a specific prediction method, and the best results can be achieved if the introduced prediction method is consistent with the final prediction method. This method has a high computational complexity and requires a well-designed feature subset search strategy to be used. The main representatives of this kind of methods are Recursive Feature Elimination (RFE), such as support vector machine recursive feature elimination (SVM-RFE) [26].

(3) Embedded method [21]. The process of feature selection is embedded in the machine learning method. Some machine learning methods can score the features during learning, such as regression model [27], SVM [28], decision tree [29], random forest [30] and so on. Finally, the features are selected by the set threshold. This kind of method does not require additional feature selection calculations, and the computational cost is relatively small, which can achieve better effects of feature selection and prediction.

(4) Ensemble method [31]. The ensemble method is a feature selection method that has emerged in recent years. Different from the above methods which directly obtain an optimal feature subset, the ensemble method obtains multiple optimal feature subsets by using the previous methods and then converges these optimal feature subsets or corresponding learning results. Due to the ensemble technology, the feature selection of the ensemble method generally has better stability than the previous feature selection methods. For example, in [31]– [32], a number of different scoring methods have been integrated to select the final feature subset and achieved good results.

Since these methods have different emphasis on feature selection, the final selected optimal feature subsets will have different to some extent. In order to make the final feature selection stable and robust, the ensemble method is generally used for feature selection. This paper is also based on this reason to choose the ensemble of multiple methods to achieve the final feature selection.

III. OUR FSPRED FRAMEWORK

This section first introduces the overview of FSPred and then describes the details of every component in this framework.

A. THE OVERVIEW OF FSPRED

Figure 1 describes the overview of our FSPred framework. First, it applies the feature transformation extraction algorithm to generate features from the training data and form an initial feature set. Then three methods are used to score the initial features respectively with mutual information, random forest, and RFE, and the obtained score results will be combined to obtain the final feature ranking set. An improved forward search method based on logistic regression is used to select the final optimal feature subset. The logistic regression model, which is easy to understand and compute, with the optimal feature subset is adopted to make predictions on test data or new examples.

B. DATA DESCRIPTION

The typical clickstream data during MOOCs learning mainly includes student registration information, the time when the behavior occurred, the object of the operation, the specific

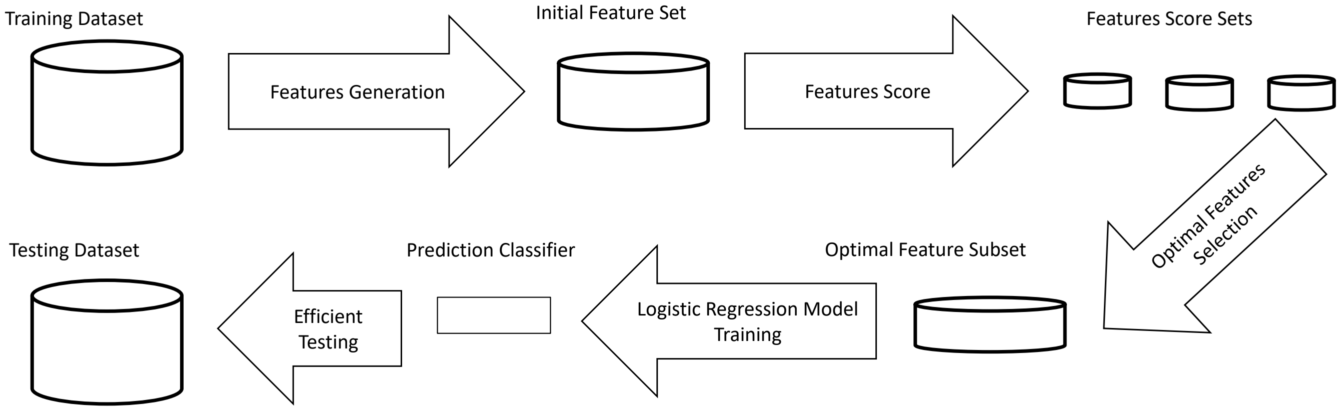

FIGURE 1. The Overview of FSPred.

TABLE 1. Attributes in Logs

Attribute	Description
Enrollment_id	The enrollment record Identification, that is associated with which student enrolls in which course.
Time	The time when the event occurs
Event	Event type
Object	The object that the student access

behavior type, etc. [33]– [34], as shown in Table 1. Generally, this kind of data containing time cannot be directly adopted by various data mining methods, so it must be converted to some extent. A common conversion method is to perform statistics on features in a certain time window. In order to minimize the loss of information details and temporal relationships caused by the transformation, we do the statistics of different events on different objects in days and combine them in time sequence. However, this method will increase the data dimension as the number of days increases, making the calculation of the latter more and more expensive. In order to reduce the overall computational overhead, feature selection methods need to be introduced to reduce redundant or invalid features.

C. FEATURE GENERATION

Since the original clickstream data contains time that cannot be used directly, it must be transformed to generate features for use by the prediction algorithm, so we first propose a fine-grained feature generation algorithm, as shown in Algorithm 1. We use the event type in the clickstream data as the category of statistics, and the learning time of all event types of all course registrants is counted in days. The reason is that the general learners usually have a continuous learning behavior in one day, which can reflect the details of learning and all the events of the days of learning can be used as features to retain the time attribute to the greatest extent. Finally, all statistical results are unified into the same length of time. This is mainly to achieve data alignment and facilitate subsequent calculation and processing.

Algorithm 1 Feature Generation Algorithm

```
Input: courses set $C$, students set $S$, event types set $T$, events set $E$
Output: the dataset $F$ with generation features
$F \leftarrow \emptyset$
for each course $c_i \in C$ do
    Get events set $E^i \in E$ of $c_i$
    Get the start time $TS_i$ and the end time $TE_i$ of $E^i$, the total number of days $D_i$ of course $c_i$
    Get students set $S^i \in S$ of $c_i$
    for each student $s^i_k \in S^i$ do
        Get events set $E^i_k \in E^i$ of $s^i_k$
        Set event count set $ET^i_k = \{ET^i_{k,m,n} = 0|m = D_i, n = count(T)\}$
        for each event $e^i_{kl} \in E^i_k$ do
            for $t = 1$ to $D_i$ do
                if time of $e^i_{kl}$ in the day $t$ and $e^i_{kl}$'s event type is $T_n$ then
                    $ET^i_{k,m,n} \leftarrow ET^i_{k,m,n} + 1$
                end if
            end for
        end for
        $F \leftarrow F \cup \{(c_i, s^i_k, ET^i_k)\}$
    end for
end for
```

D. FEATURE SELECTION

The large number of event-related statistical features obtained in the previous step may have features that are not related to classification or redundant to each other. In order to reduce the computational complexity and lower accuracy caused by such features, further feature selection is required to preserve features that are highly correlated with the classification problem.

Suppose $F = \{F_1, F_2, ..., F_N\}$ is a candidate feature set with N features. The goal of feature selection is to select a subset $S \subset F$ with k features to make the final prediction

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2881275, IEEE Access

effect optimal. The prediction effect here is related to the evaluation method actually selected, that is, the ultimate goal is to get a feature subset to maximize the value of the selected evaluation method. A simple method is to directly use the feature correlation metric to score all features, such as chi-square check, mutual information, maximum information coefficient, Pearson correlation coefficient, etc., and then select the final feature subset with the top k ranking features according to the set number of features to be selected k. This method is also called the top-k selection method. However, there are two problems to be solved in this approach. First, this method does not depend on a specific learning algorithm for feature scoring, and may not achieve stable performance in the certain learning algorithm. Second, the choice of k is often manually specified. The feature subset selected in this way may not be the true optimal subset, and thus the optimal prediction effect cannot be achieved. Therefore, we propose an ensemble feature selection method to address these two problems.

We divide the process of feature selection into two phases: feature scoring and feature search.

1) Feature scoring

We integrate three different types of methods: mutual information (MI), random forest (RF), and recursive feature elimination (RFE). Each method scores all the features independently, and then normalize the scores of each method. After averaging the scores of the same features, we rank all features by feature score in descending order.

First, the mutual information of all features F_i and class C is calculated. Mutual information is used to measure the mutual dependence between two sets of events. The mutual information of two discrete random variables X and Y is defined as in

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right). \quad (1)$$

Where $p(x,y)$ is the joint probability distribution function of X and Y, $p(x)$ and $p(y)$ are the edge probability distribution functions of X and Y respectively. In the case of continuous random variables, the sum is replaced by a double definite integral, as in (2).

$$I(X;Y) = \int_Y \int_X p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) dxdy. \quad (2)$$

Here $p(x,y)$ is currently the joint probability density function of X and Y, $p(x)$ and $p(y)$ are the edge probability density functions of X and Y respectively.

Mutual information can be used to measure the mutual dependence between a feature and a specific category when dealing with classification problem. If the amount of information is larger, the mutual dependence between the feature and the category is greater. Conversely, the mutual dependence of this feature with this category is smaller. When X and Y are independent, $p(x,y) = p(x)p(y)$, therefore

$$\log \left(\frac{p(x,y)}{p(x)p(y)} \right) = \log 1 = 0. \quad (3)$$

In order to unify the three different feature evaluation methods, we normalize the calculated $I(F_i;C)$ to obtain the score SMI_i of the feature F_i and finally obtain the score set $SMI = \{SMI_1, SMI_2, ..., SMI_N\}$.

At the same time, the random forest is run for the same training set for learning. According to the defined evaluation metrics, the optimal parameter settings are found and the important measures of all variables are obtained. Random forest is an ensemble machine learning method, which uses the random resampling technique and node random splitting technique to construct multiple decision trees and obtain the final classification result by voting. RF has the ability to analyze complex interaction classification features, is robust to noise data and data with missing values, and has a fast learning speed. Its variable importance measure can be used as a feature selection tool for high-dimensional data.

A random forest consists of multiple decision trees. Each node in the decision tree is a condition about a feature in order to split the data set into two according to different response variables. For classification problems, Gini impurity or information gain is usually used to determine the nodes. For regression problems, the variance or least squares fitting is usually used to determine the nodes. For dropout prediction in MOOCs, which belongs to the classification problem, we use Gini impurity to divide the nodes. Generally, the Gini index is usually used to calculate the purity of the node. At node t, the attribute a is used as the partition attribute to estimate the probability of belonging to different classes. The Gini index is defined as in

$$G(t) = 1 - \sum_{k=1}^{Q} p^2(k|t). \quad (4)$$

Q is the number of sample types. A larger Gini index indicates a lower purity. When the variable X_j is used as a partitioning variable, the difference between the Gini index before and after the division is calculated, represented as $\triangle_j$. Then, the impurity of the average decrease of nodes partitioned by the variable X_j in all trees is called the average decrease of Gini index, shown as $\overline{\triangle}_j$, which is used as a measure of the importance of the variable X_j. This method is also called mean decrease impurity. Similarly, we will normalize $\overline{\triangle}_j$ and get a new score for the feature F_i, and finally get the score set of all features $SRF = \{SRF_1, SRF_2, ..., SRF_N\}$.

Then, the feature is sorted and scored by introducing Recursive Feature Elimination (RFE) in the process of searching feature subsets. The main idea of RFE is to repeatedly construct predictive models with features containing weights (for example, linear models or support vector machines), here we choose logistic regression model consistent with the final prediction, and select features by recursively reducing the size of the feature set examined. First, the predictive model is trained on the original features, and each feature gets a weight. After that, those features with the minimum absolute weight are kicked out of the feature set. So recursively until all the features are traversed, as shown in Algorithm 2, the order in which features are eliminated during this process is the ordering of features.

Algorithm 2 Recursive Feature Elimination

```
Input: dataset $F$ with generation features using Algorithm
    1 and Logistic Regression model $f(\cdot)$
Output: feature ranking set $FR_{rfe}$
$F_f \leftarrow$ get the set containing features of $F$
$FR_{rfe} \leftarrow \emptyset$
repeat
    $F' \leftarrow$ extract all features of $F_f$ and the corresponding values in $F$
    Train the selected classifier with the maximization evaluation criterion to obtain the weights $Weight_i$ of all current features
    $F_{fmin} \leftarrow$ get the feature of minimum weight
    $FR_{rfe} \leftarrow F_{fmin} \cup FR_{rfe}$
    $F_f \leftarrow F_f - F_{fmin}$
until $F_f = \emptyset$
```

After the final sorted set is obtained, the features are numbered in right-to-left order and then the numbers are normalized to obtain a new set of scores $SREF = \{SREF_1, SREF_2, ..., SREF_N\}$.

Finally, we compute the average score of the three scoring sets for each feature as the final feature ranking basis and sort the scores in descending order to get the new feature ranking set $FR = \{FR_1, FR_2, ..., FR_N\}$.

2) Feature search

The feature ranking set FR obtained in the previous step is used as the input feature set, and the improved forward search method is applied for feature search until the final training effect cannot be improved. The forward search method only adds one feature with the best score currently at a time instead of traversing all the remaining features, which greatly reduces the computational complexity and improves the efficiency of the search. Each time a new feature is added, a new feature set and corresponding evaluation score are obtained. Therefore, we can turn the problem of finding the optimal feature subset into the problem of finding the optimal evaluation score under different number of features. The problem is to find an inflection point where the changes of the evaluation scores tend to be stable. In order to avoid the influence of the fluctuation of the evaluation scores on finding the inflection point that tends to be stable, we use the moving average to smooth the evaluation scores, and the calculation formula is shown as

$$F_t = \frac{A_{t-1} + A_{t-2} + A_{t-3} + ... + A_{t-n}}{n}. \tag{5}$$

Where F_t is the t-th average value and n is the moving average window size; $A_{t-1}, A_{t-2}, A_{t-3}$, and A_{t-n} represent the actual values of the previous, the first two, the first three, and the first n, respectively.

We use the obtained moving average values to judge the trend of the changes in the evaluation scores. The trend changes of the curve formed by the discrete points can be approximated by the differential changes of adjacent points. When the independent variable is uniformly changed, the change of the function $f(x)$ from x_{k-1} to x_k is called the first-order difference of the function $f(x)$ at the point x_k, which is recorded as

$$\Delta f(x_k) = f(x_k) - f(x_{k-1}). \tag{6}$$

Specifically, when the current $i - 1$ features are fixed, the algorithm adds the i-th feature to the feature subset according to the result of the feature ranking, and calculates the evaluation score of the current feature subset by linear model. Then, the m-point moving average value of the i-th evaluation score is calculated. When the difference between the obtained moving average values is continuously less than ϵ for α times, the new feature is stopped adding. Then the feature subset corresponding to the maximum value of the evaluation scores from $i - \alpha - m$ to $i - \alpha$ is found to be the optimal feature subset, and the specific details are shown in Algorithm 3. Generally, the value of m is set to 5, the value of ϵ is 0.0001, and α is 5.

Here we assume that the training set has the same distribution as the test set, so it is reasonable to use the training set to sort and select features.

E. PREDICTION

The final prediction results can be maximized when the classifier selected for recursive feature elimination and forward feature search is consistent with the final predicted classifier. In order to facilitate the interpretation of the final prediction results, we choose logistic regression as the classifier used. Once the optimal feature set is selected, for each test case, we first transform its data to extract features, map it to the space corresponding to the optimal feature subset and then use the pre-trained logistic regression model to predict the corresponding category, the details as shown in Algorithm 4. Since the number of features used is limited,, the computation of the logistic regression model is fast.

IV. EXPERIMENTS AND RESULTS

In this section, we have carried out detailed experiments based on a real dataset to demonstrate the effectiveness of our proposed framework. Firstly, the dataset, the baseline methods and evaluation metrics used in the experiments are introduced. Then the experimental results are given and analyzed.

A. DATASET

In this paper, we use the clickstream dataset provided by the MOOC platform XuetangX for KDD CUP 2015 as the experimental dataset. The dataset is marked by the second dropout definition method introduced above, which is divided

Algorithm 3 Feature Selection based on Forward

```
Input: $n$ instances $\{(x_i, y_i)\}, y_i \in \{0, 1\}$, feature ranking
    set $FR$, the number of points required for moving average $m$,
    the threshold value of difference $\epsilon$, the threshold value of times $\alpha$
Output: optimal feature subset $S$
$x^f \leftarrow$ generation features using Algorithm 1
$S \leftarrow \emptyset$
$Count \leftarrow 0$
$Position \leftarrow 0$
for $i = 1$ to $n$ do
    $S \leftarrow S \cup \{FR_i\}$
    $x' \leftarrow$ extract all features of $S$ and the corresponding values in $x^f$
    Training the selected classifier with the maximum evaluation criterion to obtain
        the evaluation score $Score_i$ corresponding to the current feature subset
    if $i >= m$ then
        $p_{Si} \leftarrow \frac{1}{m} \sum_{j=0}^{m-1} Score_{i-j}$
    end if
    if $i > m$ then
        $d_{Si} \leftarrow p_{Si} - p_{Si-1}$
        if $d_{Si} < \epsilon$ then
            $Count \leftarrow Count + 1$
            if $Count = \alpha$ then
                $Position \leftarrow$ the position corresponding to the maximum value
                    of the evaluation scores from $i - \alpha - m$ to $i - \alpha$
                break
            end if
        end if
        if $d_{Si} > \epsilon$ then
            $Count \leftarrow 0$
        end if
    end if
end for
for $i = 1$ to $Position$ do
    $S \leftarrow S \cup \{FR_i\}$
end for
```

Algorithm 4 Feature Selection based on Forward

```
Input: $n$ testing instances $(x_i)$, optimal feature subset $S$ and
    Logistic Regression model $f(\cdot)$
Output: $n$ predictions of testing instances $(\hat{y}_i)$
$x^f \leftarrow$ generation features using Algorithm 1
$x' \leftarrow$ extract all features of $S$ and the corresponding values in $x^f$
for $i = 1$ to $n$ do
    $\hat{y}_i \leftarrow f(x'_i)$
end for
```

into a training set and a test set. Since the label data corresponding to the test set is not disclosed, we only use the training set with the full labels as the experimental dataset.

TABLE 2. Statistics of dataset for the experiment

Item	Number of Item
Courses	39
Students	79,186
Enrollments	120,542
Activity events	8,157,277
Dropout	95,581
Not dropout	24,961

TABLE 3. Event types

Type	Description
problem	Do homework
video	Watch the video
access	Access other objects except the video and the job
wiki	Read the Wikipedia of the course
discussion	Forum discussion
navigate	navigate other objects except the video and the job
page_close	Close the web page

The dataset consisted of 39 courses, 79,186 students, and 120,542 students enrolled in 39 courses. Within activity logs for 30 days of each course, there were 8,157,277 records of clickstream data. According to the registration number, the learner has a record of learning activities in the next 10 days in the corresponding course. If there is no record, the student is a dropout, marked as 1, and vice versa, not a dropout and marked as 0. The final number of dropouts is 95,581, and the number of students not dropping out is 24,961. The specific statistics are shown in Table 2.

A total of seven different types of events are defined in the learning behavior log dataset provided by XuetangX. The specific names and meanings are shown in Table 3.

B. BASELINE METHODS

In order to provide a reference point for the results of FSPred framework, we use two methods that are more commonly used in existing research, namely SVM [6], Logistics Regression (LR) [8], [13]– [14] and the random forest selected in the previous feature selection as the benchmark methods.

Logistics regression (LR) is a classical linear classification method. Since the FSPred framework uses it for feature extraction and final prediction, it is used as one of the benchmarks for our comparison.

Support vector machine (SVM) is a commonly used method in classification tasks. It has a wide range of applications on various issues before the popularity of deep learning. We use an RBF kernel SVM to conduct experiments.

A random forest is a classifier that contains multiple decision trees, and the category of its output is determined by the mode number of categories output by individual trees. It is widely used because of its wide adaptability and high classification accuracy [35].

In order to facilitate comparison and analysis, we will use our proposed feature generation method to generate features from the original data, then use the generated features for the learning and prediction of the benchmark method, and use the grid search to traverse a variety of parameter combinations for SVM and random forest, the best effect parameters are determined by 10-fold cross-validation. The 10-fold cross-validation divides the dataset into ten parts and takes turns to use nine of them as training data and one as test data for training and testing. Each training and testing will produce a corresponding test result. The mean value of 10 test results is returned as the final test result, and then the stability of the test results is ensured.

C. METRICS FOR PERFORMANCE EVALUATION

It can be seen from Table 2 that the ratio of positive and negative cases in experimental datasets is not balanced, and the proportion of positive cases is as high as 80%. At this time, it is not appropriate to use the correct rate to measure the results, because the correct rate of prediction is all positive cases can reach 80%. Therefore, the precision, recall, F-score [36], ROC curve and AUC score are more suitable for the evaluation of the method in the current problem [37].

For a binary classification problem, there exists a positive class and a negative class. Then there is a confusion matrix based on the actual classes and the predicted classes of the instances, shown as follows.

Where true positive (TP) denotes the number of positive instances that are predicted to be positive, false positive (FP) denotes the number of negative instances that are predicted to be positive, true negative (TN) denotes the number of negative instances that are predicted to be negative, and false negative (FN) denotes the number of positive instances that are predicted to be negative.

According to the above confusion matrix, the precision P, recall R and F1-score F1 are defined as in

$$P = \frac{TP}{TP + FP}, \tag{7}$$

$$R = \frac{TP}{TP + FN}, \tag{8}$$

$$F1 = \frac{2TP}{2TP + FP + FN}. \tag{9}$$

Among them, precision and recall often cannot be considered at the same time. Sometimes a classifier may have high precision but a low recall or a high recall but poor precision. In order to compromise precision and recall, a new evaluation indicator F1-Score has been introduced. In general, the higher the value of F1- Score, the better the performance of the corresponding classifier. The receiver operating characteristic curve (ROC) takes the false positive rate as the x-axis and the true positive rate as the y-axis. By adjusting the threshold of the classification to obtain an ROC curve, the area under the curve namely AUC can be used to evaluate the performance of a classifier. The larger the value of the AUC score, the better the performance of the classification. One of the most important characteristics of AUC score is that it is not affected by the class ratio of data samples, and it is especially suitable to be used as an evaluation metric for classification of class-imbalanced dataset [38].

TABLE 4. The information of generated features

Feature Name	Description
Coursesid	Course identifier
username	User name
i-access	Number of access events on day $i, i >= 1andi <= 30$
i-discussion	Number of discussion events on day $i, i >= 1andi <= 30$
i-navigate	Number of navigating events on day $i, i >= 1andi <= 30$
i-page_close	Number of page_close events $i, i >= 1andi <= 30$
i-problem	Number of problem events on day $i, i >= 1andi <= 30$
i-video	Number of video events on day $i, i >= 1andi <= 30$
i-wiki	Number of Wiki events on day $i, i >= 1andi <= 30$

D. EXPERIMENTAL RESULTS

We use the Algorithm 1 to convert the original data and generate a total of 212 features as the input features of all the methods. These features mainly count the number of occurrences of each event in days, as shown in Table 4.

The generated features are fed into the FSPred framework and benchmark methods as input, and the respective predicted evaluation results are shown in Table 5. All methods use 10-fold cross-validation to obtain the final generalization assessment results. Among them, the parameters of SVM and random forest are determined by grid search. The parameter C of SVM (with RBF kernel) is 1 and the value of $gamma$ is 1/212. The training process parameters of the random forest are selected step by step by grid search. During this period, for each selected parameter value, 10-fold cross-validation is used to obtain a mean value of AUC, and finally the parameter value with the highest AUC is selected as the value of the corresponding parameter. Firstly, we look for the optimal number of decision trees. The search range is from 100 to 300 with a step size of 20, and the maximum number of features of each subtree is the square root of the number of all features. The optimal number of decision trees is determined to be 280. Next, we search the maximum depth of each decision tree and the minimum number of samples needed to divide the internal nodes. The search range of maximum depth is from 3 to 14 with a step size of 2, the search range of minimum sample number is from 50 to 200 with a step size of 20. Then the final maximum depth is 13, and the minimum number of samples is 50.

Comparing the first row and the second row in Table 5, it can be seen that the prediction results of the features extracted by our method are slightly better than those by the manual method in the literature [36] on the three evaluation indicators, which shows that our feature generation method is effective. From the second row to the fifth row, the final accuracy of the prediction using the FSPred framework is basically the same as the benchmark method. The F1 Score and the AUC are slightly worse than the random forest, but considering that the number of features we use is only 79, which is 133 fewer than the 212 features used in the benchmark method. Especially the FSPred framework only needs 79 features to reach the same level as the benchmark LR, so the computational overhead will be much less when new data arrives. In our experiments, the average predicted execution

TABLE 5. Results of the experiments in different algorithms with all courses

Algorithms	Precision	Recall	F1 Score	AUC
LR in [36]	0.844*	0.852*	0.843*	-*
LR	0.8567	0.8629	0.8469	0.8630
SVM(with RBF kernel)	0.8400	0.8307	0.7856	0.8604
Random forest	0.8524	0.8571	0.8543	0.8647
FSPred with 79 features	0.8573	0.8631	0.8469	0.8629
LR with 143 features using MI	0.8562	0.8623	0.8459	0.8629
LR with 137 features using RF	0.8573	0.8633	0.8472	0.8629
LR with 105 features using RFE	0.8565	0.8627	0.8466	0.8629

The data with * is from the calculation results in [36], which does not include AUC, so the corresponding position is -.

time of FSPred is 7 milliseconds, which is close to 1/4 of the LR's average prediction time of 26 milliseconds, much less than the random forest's 174 milliseconds and SVM's more than 4 minutes, and it is easy to extend to online real-time forecasting. Rows 5 to 8 of Table 5 compare the effects of the single feature selection algorithm and the integrated feature selection algorithm in FSPred on the results. Although the feature subsets selected by these methods ultimately achieve a similar effect, the integrated method can find a better feature subset (fewer features) than a single method. It should be noted that the execution time of our integrated feature selection method is much longer than the time required for feature selection using mutual information, random forest and RFE alone, which is slightly less than the sum of the time of the three methods. In practice, it is more suitable for offline feature selection and model training. The computational cost after determining the feature subset and model is much less than the single method.

Figure 2 visualizes the changes in Precision, Recall, F1 Score, and AUC values of the FSPred framework under different numbers of features. It can be seen that after the introduction of the first 79 features, the values of Precision, Recall, F1 Scores and AUC have tended to be stable and their fluctuations are small. This is consistent with the feature selection results in Table 5. It is indicated that there are a certain amount of redundant features or noise features in the generated high-dimensional features, and the feature selection method in FSPred framework can effectively remove these features.

Figure 3 shows the type distribution of the features in the optimal feature subset. The feature type 'Username' is the corresponding feature in Table 4, the count of which is 1. The other feature type names correspond one-to-one with the event types in Table 3 and each feature type name indicates the selected features generated by the same related event. Since the dataset has only 30 days of each course, the feature count under the feature types is a maximum of 30. For example, 'Access' indicates the selected features of different days generated by the access event, and the number is 22. As can be seen from this figure that the student dropout and the student's behavioral events are closely related, and different courses and students will bring different results. Here we can also see that the features of the event 'Discussion' and 'Wiki' do not appear in the optimal feature subset, indicating that the event 'Discussion' and 'Wiki' have the least impact on the prediction result, and have almost no impact. One possible reason is that the role of the discussion and wiki modules in the MOOC platform XuetangX is not reflected. The discussions and wikis of many courses are blank or seldom content, so they do not play a role in promoting learning. This situation requires the attention of MOOC platform managers and teachers.

V. CONCLUSION

In this paper, we present an effective prediction framework based on ensemble feature selection (FSPred) to solve the problem of dropout prediction in MOOCs. FSPred first converts the students' clickstream log data and extracts the detail features in days, scores the input features through the ensemble feature selection method, and then uses the forward search method combined with the prediction effect of LR, which is easy to calculate and interpret, to find the optimal feature subset. Finally, we use the obtained feature subset and the trained LR to predict the new data. Experiments on open datasets and selected benchmark methods show that FSPred has the ability to extract features and find valid features for MOOC students' clickstream data to improve predictive performance and reduce computational complexity. Finally, the main factors affecting dropout are analyzed by the selected features, which provides a certain reference for the construction of MOOCs.

ACKNOWLEDGMENT

We thank the organizers of KDD Cup 2015 as well as XuetangX for making the datasets available.

REFERENCES

[1] D. Clow, "MOOCs and the funnel of participation," in Proceedings of the Third International Conference on Learning Analytics and Knowledge. ACM, 2013, pp. 185–189.

[2] Y. Wang, "Exploring possible reasons behind low student retention rates of massive online open courses: A comparative case study from a social cognitive perspective," in Proceedings of the 1st Workshop on Massive Open Online Courses at the 16th Annual Conference on Artificial Intelligence in Education, 2013, p. 58.

[3] G. Balakrishnan and D. Coetzee, "Predicting student retention in massive open online courses using hidden markov models," Electrical Engineering and Computer Sciences University of California at Berkeley, p. 13, 2013.

[4] R. F. Kizilcec, C. Piech, and E. Schneider, "Deconstructing disengagement: analyzing learner subpopulations in massive open online courses," in Proceedings of the third international conference on learning analytics and knowledge. ACM, 2013, pp. 170–179.

[5] S. Halawa, D. Greene, and J. Mitchell, "Dropout prediction in MOOCs using learner activity features," Experiences and best practices in and around MOOCs, vol. 7, pp. 3–12, 2014.

[6] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart, "Predicting MOOC dropout over weeks using machine learning methods," in Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs, 2014, pp. 60–65.

[7] D. F. O. Onah, J. E. Sinclair, and R. Boyatt, "Dropout rates of massive open online courses: Behavioural patterns," in International Conference on Education and New Learning Technologies, 2014, pp. 5825–5834.

[8] C. Taylor, K. Veeramachaneni, and U.-M. O'Reilly, "Likely to stop? predicting stopout in massive open online courses," arXiv preprint arXiv:1408.3382, 2014.

[9] D. S. Chaplot, E. Rhim, and J. Kim, "Predicting student attrition in MOOCs using sentiment analysis and neural networks." in Proceedings of

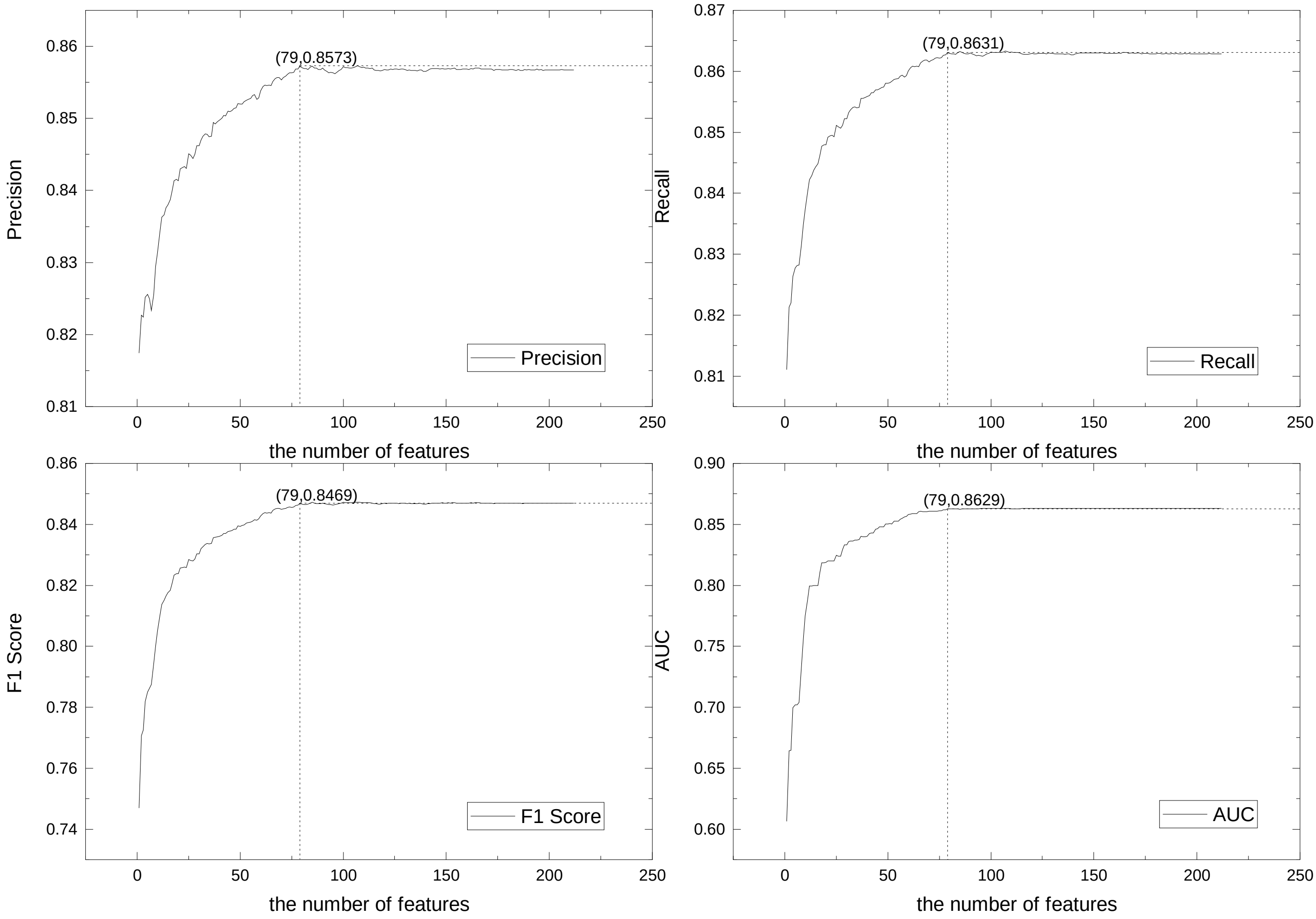

FIGURE 2. Precision, Recall, F1 Score and AUC corresponding to different number of features.

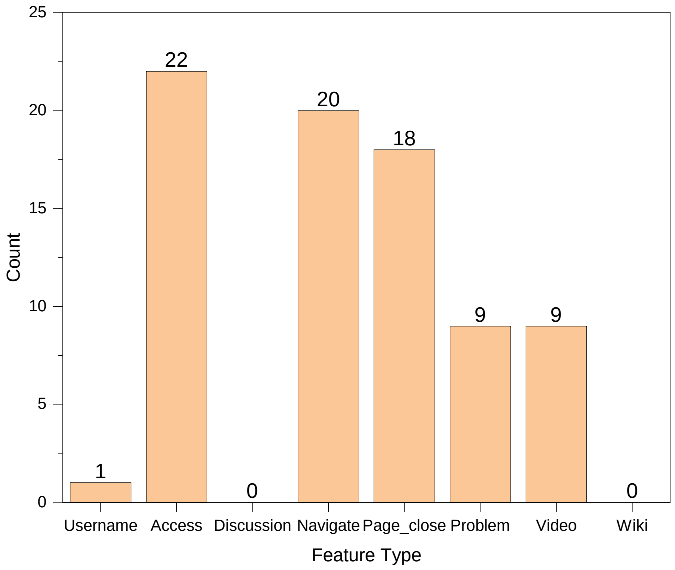

FIGURE 3. The type distribution of the features in the optimal feature subset.

the 2015 AIED Workshop on Intelligent Support for Learning in Groups, 2015, pp. 7–12.

[10] M. Fei and D.-Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," in Data Mining Workshop (ICDMW), 2015 IEEE International Conference on. IEEE, 2015, pp. 256–263.

[11] J. L. Hung, M. C. Wang, S. Wang, M. Abdelrasoul, Y. Li, and W. He, "Identifying at-risk students for early interventions #x2014;a time-series clustering approach," IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 1, pp. 45–55, 2017.

[12] R. M. Stein and G. Allione, "Mass attrition: An analysis of drop out from a principles of microeconomics MOOC," Social Science Research Network, pp. 1–19, 2014.

[13] S. Jiang, A. Williams, K. Schenke, M. Warschauer, and D. O'dowd, "Predicting MOOC performance with week 1 behavior," in Proceedings of the 7th International Conference on Educational Data Mining, 2014.

[14] J. He, J. Bailey, B. I. Rubinstein, and R. Zhang, "Identifying at-risk students in massive open online courses." in Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015, pp. 1749–1755.

[15] F. Wang and L. Chen, "A nonlinear state space model for identifying at-risk students in open online courses." in Proceedings of the 9th international conference on educational data mining, 2016, pp. 527–532.

[16] W. Xing, X. Chen, J. Stein, and M. Marcinkowski, "Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization," Computers in Human Behavior, vol. 58, pp. 119–129, 2016.

[17] W. Wang, H. Yu, and C. Miao, "Deep model for dropout prediction in MOOCs," in Proceedings of the 2nd International Conference on Crowd Science and Engineering. ACM, 2017, pp. 26–32.

[18] J. De Weerdt, "Dropout prediction in MOOCs: A comparison between process and sequence mining," in Business Process Management Workshops: BPM 2017 International Workshops, Barcelona, Spain, September 10-11, 2017, Revised Papers, vol. 308. Springer, 2018, p. 243.

[19] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley, "MOOC dropout prediction: How to measure accuracy?" in Proceedings of the Fourth (2017) ACM Conference on Learning at Scale. ACM, 2017, pp. 161–164.

[20] Y. Chai, C.-U. Lei, X. Hu, and Y.-K. Kwok, "WPSS: dropout prediction for MOOCs using course progress normalization and subset selection," in Proceedings of the Fifth Annual ACM Conference on Learning at Scale. ACM, 2018, p. 29.

[21] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of machine learning research, vol. 3, pp. 1157–1182, 2003.

[22] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in Tools with artificial intelligence, 1995. proceedings., seventh international conference on. IEEE, 1995, pp. 388–391.

[23] N. Hoque, D. K. Bhattacharyya, and J. Kalita, "MIFS-ND: A mutual information-based feature selection method," Expert Systems with Applications, vol. 41, pp. 6371–6385, 2014.

[24] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 856–863.

[25] R. Kohavi and G. H. John, "Wrappers for feature subset selection," Artificial intelligence, vol. 97, no. 1, pp. 273–324, 1997.

[26] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Machine learning, vol. 46, no. 1, pp. 389–422, 2002.

[27] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in Proceedings of the twenty-first international conference on Machine learning. ACM, 2004, p. 78.

[28] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines." in ICML, vol. 98, 1998, pp. 82–90.

[29] V. Sugumaran, V. Muralidharan, and K. Ramachandran, "Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing," Mechanical systems and signal processing, vol. 21, no. 2, pp. 930–942, 2007.

[30] V. Svetnik, A. Liaw, and C. Tong, "Variable selection in random forest with application to quantitative structure-activity relationship," Proceedings of the 7th Course on Ensemble Methods for Learning Machines, 2004.

[31] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2008, pp. 313–325.

[32] J. Xu, L. Sun, Y. Gao, and T. Xu, "An ensemble feature selection technique for cancer recognition," Bio-medical materials and engineering, vol. 24, no. 1, pp. 1001–1008, 2014.

[33] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. T. Seaton, "Studying learning in the worldwide classroom: Research into edX's first MOOC," Research & Practice in Assessment, vol. 8, pp. 13–25, 2013.

[34] K. Veeramachaneni, S. Halawa, F. Dernoncourt, U.-M. O'Reilly, C. Taylor, and C. Do, "Moocdb: Developing standards and systems to support MOOC data science," arXiv preprint arXiv:1406.2015, 2014.

[35] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[36] W. Li, M. Gao, H. Li, Q. Xiong, J. Wen, and Z. Wu, "Dropout prediction in MOOCs using behavior features and multi-view semi-supervised learning," in Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE, 2016, pp. 3130–3137.

[37] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 233–240.

[38] T. Fawcett, "An introduction to ROC analysis," Pattern recognition letters, vol. 27, no. 8, pp. 861–874, 2006.

• • •