JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 139-155 (2021)
DOI: 10.6688/JISE.202101_37(1).0010

# Adopting Software Product Lines to Implement an Efficient Learning Analytics Framework in MOOCs

CHEN-HSIANG YU[1], JUNGPIN WU[2], MING-CHI LIU[1,+] AND AN-CHI LIU[1]
[1]*Department of Information Engineering and Computer Science*
[2]*Department of Statistics*
*Feng Chia University*
*Taichung, 407 Taiwan*
*E-mail: {chyu; cwu; mingcliu[+]; acliu}@fcu.edu.tw*

Massive open online courses (MOOCs) have become very popular in education and learning analytics, and can help students understand their learning situations and assist teachers in class management. However, facing different course objectives, multiple learning activities and student's diversity in motivation, learning analytics often suffers from high complexity and inefficient data analysis. This results in a long process from the implementation of data analysis to decision support, and an inability to offer the instantaneity required by teachers. In particular, this problem in MOOC courses in schools makes it more difficult for teachers to understand students' learning situations, provide timely assistance, and improve course pass rates. Therefore, how to use a good analytics framework to quickly establish various analysis models with convenience and flexibility is of particular importance in MOOC development.

Current data analytics frameworks only focus on the provision of data analysis steps, and fail to consider the variability of data analysis and the repeatability of analysis results in terms of similar problems. This study attempts to apply the concept of Software Product Lines (SPL) in software engineering technology to the framework of data analytics. SPL can guide users and make the data analysis process more reusable, just like the development of software products. To verify the feasibility and effectiveness of the proposed framework, this study built practical machine learning models on the framework to predict learning performance through student learning behavior. The results show that the SPL-based approach can be used to build effective MOOC learning analytics frameworks.

***Keywords:*** MOOCs, learning analysis, software product lines, framework, machine learning

## 1. INTRODUCTION

Many innovative learning models have arisen in the education field in recent years, including various online learning platforms, which are conducive to the accumulation of a large number of learning data, and learning analytics can help students understand their learning status and assist teachers in class management. In particular, there is a growing use of Massive Open Online Courses (MOOCs) [1, 2] in education today, and the Ministry of Education in Taiwan has promoted MOOC programs for universities since 2014. A total of 63 colleges and universities have participated in this program, 341 courses have been launched, and more than 500,000 students have registered. However, the low course completion rate of MOOC courses is a problem of particular concern to educators. As a result, considerable research has focused on the use of learning analytics to help improve course completion rates.

Received November 19, 2018; revised March 2, 2019; accepted June 4, 2019.
Communicated by Hung-Yu Kao.
[+] Corresponding author.

Learning Analytics uses learning process records to analyze students' learning data, and to monitor and understand their learning behavior. The purpose is to understand a learner's learning performance, and to improve the learning environment and outcome. This can provide learners, teachers, and schools with feedback that can be applied to understanding the learner's progress, offering them tutorship catering to their individual learning needs, and allow teachers use it as a basis for adjusting their teaching contents in order to improve learning results. However, different teaching objectives of different courses, the diversity of learning activity design and the differences between students in a course often increase the complexity and inefficiency of learning analytics.

When various learning analysis platforms are developed, the software is usually developed in terms of one research topic or a specific function. The reuse of the core data set and calculation components are rarely considered, which results in a lack of flexibility during modification, meaning development must start from scratch almost every time. This means that, since the processing efficiency of vast amounts of data is critical, when a new efficient algorithm appears, it must be used in the original application, or a new application must be developed, which precludes the advantages of reuse of the components.

However, previous software process models, including Waterfall, Prototyping, Spiral, Object-oriented, Agile, and other incremental or iterative approaches, are not suitable for solving the above problems, and the control cost and requirement compromises paid by re-oriented software engineering on component analysis and requirement modification cannot meet the needs of this study [3]. Therefore, this study focused on the Software Product Lines (SPL) [4] approach, and found that SPL could reuse components with similar functions and adjust software components based on users' requirements to take advantage of reuse to improve system quality, reduce cost, and speed up the development of an application system.

This study therefore proposes a learning analytics framework based on the Software Product Lines approach, and constructed MOOC data analysis architecture with open source programs under the cluster computing environment. Therefore, learners, teachers and administrators can independently choose the core assets data set to be presented through the analysis framework based on personal needs, and show learning activity indicators of the courses, which can be used as the basis for changes to improve learning outcomes. They can also make use of the framework architecture and core assets to develop other application systems, such as the development of personalized courses, active learning, and other customized systems.

## 2. LITERATURE REVIEW

### 2.1 MOOCs

Currently, the most popular MOOC platforms in the world include **OpenEdX** [3] jointly established by Massachusetts Institute of Technology, Harvard University and UC Berkeley, **Coursera** [5] founded by two professors in Information Engineering from Stanford University, the **Khan Academy** founded by Salman Khan, a graduate of Massachusetts Institute of Technology and Harvard University, and **Udacity** [6] funded by Sebastian Thrun, David Stavens and Mike Sokolsky. All of these prestigious organizations offer hundreds of free courses, allowing anyone to access the course resources and interact

with other peers via the Internet, while provide opportunities to interact with course teachers or assistants.

As MOOCs are a kind of personalized autonomous learning, in order to help learners improve their learning results, many researchers have focused on the analysis of the learning process records of MOOC users [7-9]. As a result, they can predict the performances of students and provide assistance as needed. Therefore, these platforms also focus on continuous evaluation and improvement of the learning experience in the field of digital learning.

### 2.2 Learning Analytics

An emerging research field, learning analytics' main research focus is on learners, by collecting and analyzing related learning data and then evaluating learning results or optimizing the learning process and environment. User learning process records are generated through the system's automatic capture of the interactive data of an online platform. Chatti *et al.* proposed the reference model of learning analytics in 2013 based on four dimensions, namely What (data, environment and context), Who (stakeholders), Why (objectives) and How (methods) [10-12]. To evaluate users' learning behavior and achievements, their video watching activities and test results can be analyzed. Most learning platforms monitor and record the whole learning process in various logs. The results of learning analytics can provide users with their learning status and performance level, making them aware of their problems or areas that need further work in order to improve their results. On the other hand, teachers can use the analysis results to see if the learning outcomes are as expected, or if modification of teaching activities and course materials is required. The interaction data between users as well as between users and teachers are valuable resources for learning analytics, allowing better understanding and better communications among the platform stakeholders.

### 2.3 Software Development Model and Software Product Lines

The software development model refers to the whole process of software development, activities and the structure and records of the related tasks, including the requirement development, design, program writing, testing, deployment and maintenance phases. Commonly used software development models include Waterfall, Agile, Object-oriented, Software Product Lines [13, 14], *etc.* The Waterfall model divides the life cycle of software into the six essential activities of planning, requirement analysis, design, programming, software testing, and operation maintenance. These activities are in fixed order from the top down, just like a waterfall, which means the model lacks flexibility. Although the Agile model is relatively flexible, and manages the development of products more effectively through incremental and iterative processes, it is no better than waterfall in terms of reuse. Object-oriented programming is a programming method which uses the concept of objects [15]. The object is used as the basic unit of the program, and the program and data are encapsulated in the object to improve the reusability, flexibility, and expandability of the software. The Object-oriented model is suitable for the reuse of objects and encodings.

Software Product Line development establishes core assets, and then develops similar software systems based on those assets with different properties in terms of specific fields (see Fig. 1). The core of SPL is strategic reuse, and it can reuse various types of software

components in different software development stages, thereby improving the reuse rate of software components [16]. Compared to Object-oriented programming, SPL is more suitable for reuse, and more flexible in terms of the overall software development process. Its primary process consists of two major phases. The first phase is called domain engineering, in which core assets which can meet general demands are developed. The second phase is called application engineering, in which the core assets are reused to develop products that meet customers' specific requirements. Therefore, the Software Product Line approach is based on the practice of reusing existing software assets as far as possible, and then developing a series of similar products that meet the requirements of different users. In addition, core assets can be established and managed in specific fields. When new product development takes place, core assets can be used to integrate the components developed for new requirements for the best overall benefit.

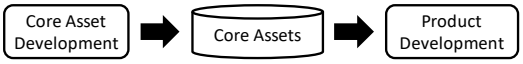


Fig. 1. Software production line process.

### 2.4 Machine Learning

Machine learning classifies collected data and trains prediction models using algorithms, in order to make predictions using a trained model when new information is obtained. Machine learning techniques like Naive Bayes, Random Forest, Decision Tree and SVM (Support Vector Machine) can be used to predict the performance of students, which can help instructors to improve their course design accordingly [17]. For example, [2] used SVM, Logistics Regression, Random Forest and Gradient Boosting Decision Tree machine learning to make dropout predictions, while [18] employed two types of neural network, Feedforward Neural Network (FFNN) and Self Organized Map (SOM), to predict whether learners would receive certifications at the end of their courses. The data used in machine learning consists of feature data and real categories in the model training process. For example, in this study the K Nearest Neighbor (KNN) algorithm [19] is used for the classification of data, where K is a constant used to denote the number of points in the closest distance to determine in which category a subject belongs. Next, the SVM algorithm is used for the supervised learning model, which is commonly used for pattern recognition, classification and regression analysis. Third is the Artificial Neural Network (ANN) [20], which is composed of many nerve cell nodes. These nerve cells construct a network model composed of an input layer, an output layer and a number of hidden layers. The output result has two states, yes or no. The traditional artificial neural network can train a model through back propagation, yielding a better model for solving the problem more efficiently.

## 3. SPL-BASED MOOC LEARNING ANALYTICS FRAMEWORK

The components of the proposed MOOC learning analytics framework are described in this section. Since every MOOC platform shares some common requirements with others, and commonalities exist between the teaching objectives of some courses, it is

possible to group these conditions or capabilities as general requirements that are highly likely to be reused. As for different MOOC platforms and other courses, various specific needs or goals are treated as specific requirements. For example, recording events and predicting learning performance are general requirements, since their modules are core assets in the proposed learning analytics framework. However, they can be modified or re-built if specific needs arise for different courses, or special teaching objectives. Examples of specific requirements would be a particular type of radar chart to show students' performance, or a unique file format converter for a platform.

Fig. 2 shows the proposed MOOC learning analytics framework. This framework is divided into two parts as shown by the dotted boxes, according to the software product line method: (A) Domain engineering on the left, which targets the development of reusable core assets and aims to meet general requirements; (B) Application engineering aims to develop products that meet special needs through the reuse of core assets. This process continuously feeds back to domain engineering to ensure adequate maintenance of core assets.

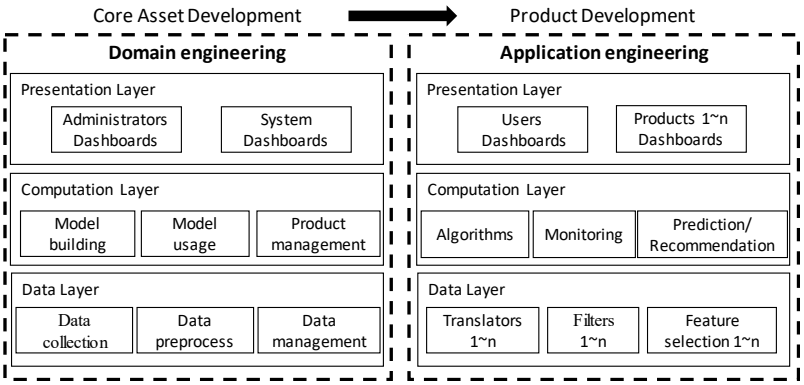


Fig. 2. SPL-based MOOC learning analytics framework.

### 3.1 Core Assets Development

#### 3.1.1 Domain engineering

Learning analytics domain engineering analysis results can be used to understand how much learners participate in a course, and how much they know, which can provide information that will enable teachers improve teaching methods. As a core asset, learning analytics uses data and models to predict the performance and progress of students, and take appropriate action. Teachers provide online courses on the learning platform, including handouts, videos, and tests. Peer students can discuss the course on the platform, and the teacher can determine students' learning states through their behaviors, and offer guidance and assistance. The learning analytics data model presents data relationships, allowing teachers to plan courses, while learners engage in various behaviors on the learning platform. These behaviors include videos watched, lecture notes, tests and discussion. Each type of behavior has entities, which have their own properties. Learning performance can thus be observed through the physical properties of different behaviors on the learning platform.

### 3.1.2 Three layers of design of learning analytics in domain engineering

This study divides the design of learning analytics into three layers, including the Data Layer, the Computation Layer, and the Presentation Layer.

*The Data layer* first engages in **data collection**, which includes the viewing of course videos, quizzes, and the recording, collection, and storage of learning activities including discussion in the discussion forum. **Data preprocessing** prepares and normalizes the collected data and transforms unstructured records into structured data as needed. **Data management** is the general management of data.

*The Computation layer* processes and analyzes data according to the objectives, including **model building**, method and library usage to construct multiple analysis models. **Model usage** refers to the use of various models to meet users' needs. **Product management** manages all the finished products, including core assets and application products. For example, the predictive models built in this study are finished products that can be reused.

*The Presentation layer* presents the analysis results in visual aids, allowing course teachers to understand a learner's status and prediction information. When specific signals are found, advice and feedback are provided to the learner to make improvements. Due to the demands of different presentations, this layer also provides **Administrator Dashboards** and **System Dashboards**.

### 3.1.3 Development of feature functions in domain engineering

A Feature Model is established in the learning analysis process. First, it includes the planning of course contents, syllabus, handouts, videos and tests. Next, course learning activities are the results of registration management, course browsing, video viewing, quiz taking, and discussion. Finally, performance evaluation examines the learning outcome of the learner. Software modules or components can then be managed using the Feature Model [14].

Using the Feature Model concept, this study describes the core assets and product development of software functions in the Data and Computation Layers, as shown in Figs. 3 and 4. The Feature Model is an abstract concept that describes the commonalities and variability of software. In this tree structure, the "feature" is the node of the tree, and the "line" is the relationship between the node and the parent node [23-25]. The commonality becomes a condition of the core assets and can be reused. Feature functions are Feature Models that are presented in terms of functions. In Figs. 3 and 4, rectangles represent functional feature items, and lines represent the different relationships between the layers. For example, the "Mandatory" relationship is shown in solid lines, indicating that the feature "Translators" must contain the feature "CourseLogTrans." The "Requires" relationship is presented in dashed lines, indicating that the presence of the feature "CourseLogSequence" depends on the feature "CourseLogTrans."

There are two feature functions in the data collection process, namely data connection and data reading from JSON, MySQL and MongoDB. There are also two feature functions in the data preprocessing process. JSON processing converts unstructured data into structured data, including the process of six video play events and the problem_check event.

MySQL processing retrieves learners' data, course registration data, course unit data, pass or fail tags and other records. Data management manages general data processing. In the Computation Layer, the model building process includes statistics and machine learning algorithms. Model usage contains two feature functions, including development environment and languages. Product management manages the built models for product development.

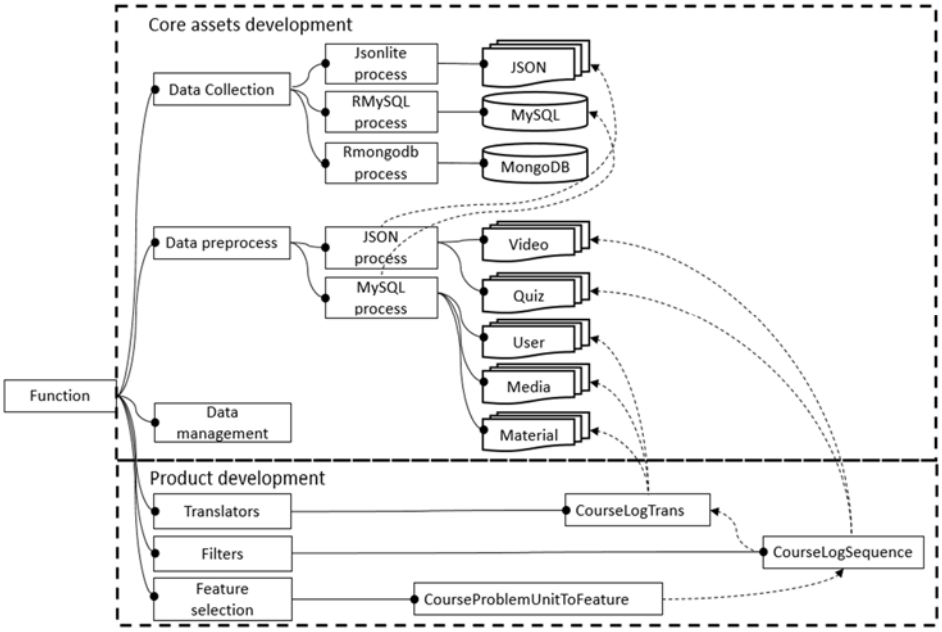


Fig. 3. Core assets and product development in the data layer.

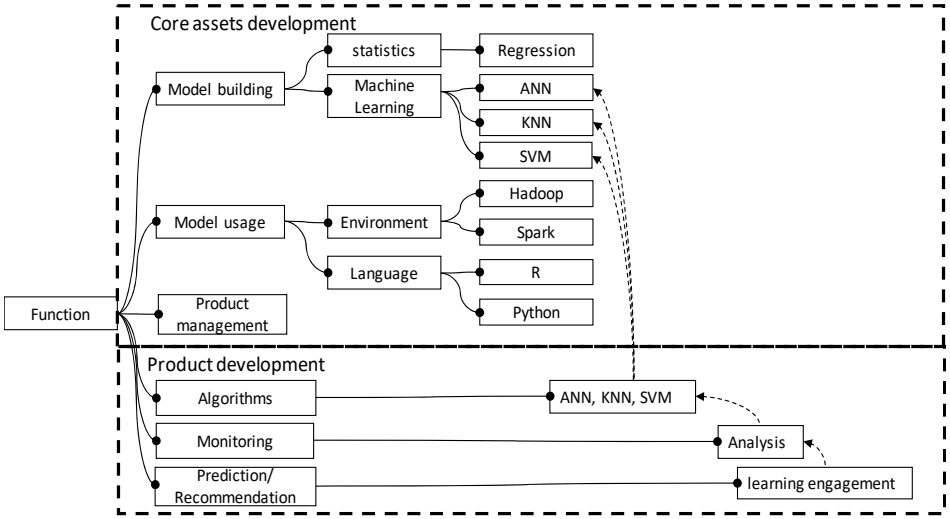


Fig. 4. Core assets and product development in the computation layer.

### 3.2 Product Development

Product development reuses core assets and develops user-specific software products. Based on the criteria for reusing the core assets, the product manager will provide developers with the necessary information to meet their general requirements. Future work will include the provision of registration and search functions to better manage the core assets for developers.

#### 3.2.1 Application engineering

Application engineering involves product development that meets specific requirements. To evaluate the learning engagement of users, this study observed their course video viewing behaviors based on the flow of video play events, as shown in Fig. 5. The load_video event was triggered when a video was completely loaded to be played. The play_video event was triggered when the play button of videos was selected. The pause_video event was triggered when the pause button was selected. The seek_video event was triggered when the video was played and different segments of the video were viewed. The speed_change_video event was triggered when the video was played at different playback speeds. The stop_video event was triggered at the end of video play.

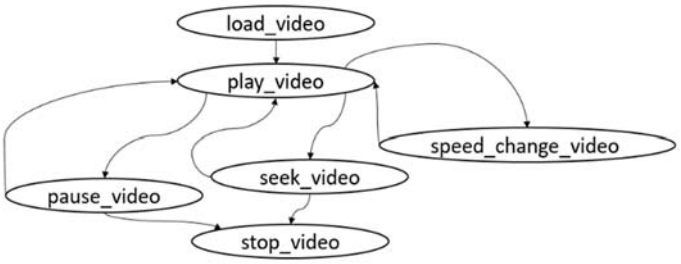


Fig. 5. Flow chart of video play events [22].

Since the target MOOC platform in this research is OpenEdu [21], the data of the platform was stored in MySQL and MongoDB, and the Tracking Log was stored in JSON format. The MySQL database contained personal user data, course learning record and basic data of the courses. The MongoDB database contained the contents of the course discussion, course videos, and course exercises. The Tracking Log recorded user operation behavior, and the content was divided into timestamped events. The events included video playing events, discussion area events, response events, and website browsing events.

This study also analyzed learning engagement in terms of the event logs produced by taking quizzes or tests. These data sets were called problem_check. Each learner took the test in each course unit. The log recorded how many tests were taken, how many times a test was tried, the score assigned to a test, the score of a test, *etc.*

#### 3.2.2 Three layers of design of learning analytics in application engineering

The Data Layer contains the translation, filters and feature selection processes.

**Translators** (CourseLogTrans) obtains data from the data preprocessing to extract the records with a specified feature set for the target courses and six video play event types.

**The filtering** function in the Data Layer (CourseLogSequence) filters data produced from the conversion function to get a meaningful set of data based on video viewing and the quiz outcome, as an example.

**The feature selection** function (CourseProblemUnitToFeature) first combines the feature sets of the video viewing and quiz outcome produced by the filtering function. Here, the generated activity feature table of the course unit has 16 features, as shown in Table 1. These 16 features are selected based on a common set of attributes that supports the analysis of students' learning behaviors and performance with respect to the teaching objectives of general MOOC courses [26, 27]. After using feature extraction to choose a proper set of features from Table 1 for a specified objective, the proposed method performs feature selection to find the best feature sets for prediction model building.

**Table 1. Feature table of course unit activity.**

| No. | Name | Descriptions |
|---|---|---|
| 1 | unit_num | Total number of course units |
| 2 | video_num | Total video number of course units |
| 3 | sess_num | Total number of online video viewing |
| 4 | load_num | Total number of video viewing by clicking load_video event |
| 5 | play_num | Total number of video viewing by clicking play_video event |
| 6 | pause_num | Total number of video viewing by clicking pause_video event |
| 7 | stop_num | Total number of video viewing by clicking stop_video event |
| 8 | seek_num | Total number of video viewing by clicking seek_video event |
| 9 | speed_change_num | Total number of video viewing by clicking speed_change_video event |
| 10 | exam_num | Total number of tests of units |
| 11 | prom_num | Total number of times of taking tests |
| 12 | all_attempts | Total number of times of trying tests |
| 13 | unit_score | Total scores of correct answers of unit tests |
| 14 | final_score | Total scores of correct answers of final test |
| 15 | final_result | Final scores of passing the course |
| 16 | total_score | unit_score ∗ 0.4 + final_score ∗ 0.6 |

The Computation Layer of product development includes algorithms, monitoring and Prediction/Recommendation. The **monitoring** function examines and adjusts the model accuracy based on the algorithm results. The **prediction and recommendation** functions make predictions and recommendations based on the generated model under the monitoring function. The User Dashboards and Product Dashboards comprise the presentation layer of the application engineering process.

### 3.2.3 Development of feature functions in application engineering

The Data Layer contains three feature functions. The Translators part has a CourseLogTrans function to convert OpenEdu learning activity records into structured records.

The Filters part has a CourseLogSequence function to convert the structured record of the course into a chronological event record. The Feature Selection part has a CourseProblemUnitToFeature function to convert a chronological event record into a unit's event record.

In this study's implementation, the Computation Layer also contains three feature functions. The algorithms use ANN, KNN and SVM for the performance prediction function in the course. The monitoring includes an Analysis function to evaluate the levels of course participation using the predictive model. The prediction and recommendation element include the learning engagement function based on the Analysis results. This function can produce prediction results and recommend a list of students for further instruction.

## 4. SYSTEM IMPLEMENTATION AND EXPERIMENTS

To verify that the proposed SPL-based Analytics framework is feasible, this study implemented a machine learning model to predict learning effect using the learning behaviors of course videos watched and tests taken on the OpenEdu platform. The model acted as the development result of core assets, and it is used to assist product development in application systems. This study's implementation environment is shown in Table 2, open source tools were used for development, and the function set used is shown in Table 3.

**Table 2. Experiment environment.**

| Operating System | CentOS 7 |
|---|---|
| CPU | Intel(R) Core(TM) i5-4570 |
| CPU Frequency | 3.20GHz |
| RAM Size | 16GB |
| Program Language | R-3.35.0 |
| Development Tools | RStudio |
| Database | MySQL |

**Table 3. Function set list.**

| Name | Command | Description |
|---|---|---|
| nnet | ann | Feed-Forward Neural Networks and Multinomial Log-Linear Models |
| ISLR | knn | *k*-Nearest Neighbor Classification |
| e1071 | svm | Misc Functions of the Department of Statistics, Probability Theory Group |
| caret | findCorrelation | Classification and Regression Training |
| Hmisc | rcorr | Matrix of Correlations and P-values |
| stats | cor | Correlation, Variance and Covariance |
| RMySQL | dbConnect<br>dbDisconnect | Database Interface and 'MySQL' Driver for R |

This study chose a physics experiments foundation course from OpenEdu as the subject in this research. The course consists of theoretical concepts, experiment demonstrations, and data analysis. The period of this course is six weeks, starting from 2014/12/1 to 2015/4/12. The teaching materials included 22 units, 21 tests, 55 videos and 532,579

learning records. Each course unit contained 1 to 4 videos, and 0 to 2 tests. Each test had 1 to 3 questions. A total of 1,387 students were registered for the course, 40 students dropped the course, and 1,258 students enrolled successfully; 590 students completed the course, and 264 students obtained the certificate, while 326 students failed to obtain the certificate.

The development of core assets is described first. The course data for the whole period was used to establish the prediction model using machine learning, and this was used as the core assets. The product development used core assets and learning activity records in the new period to make pass/fail predictions, and provide information to help students that may need the tutorship on a weekly basis. Since this study did not have the course data for the new period, the current data was used as an example to illustrate the applicability of the proposed approach.

### 4.1 Development of Core Assets

The development of core assets first involves the analysis of the predictability between learning activity and learning effect in order to establish the prediction model. First, the absolute value of a student's course grade (final_result) is converted to a binary classification of passing (1) and failing (0), that is, whether the student pass a course is used as the prediction objective. Other features of learning activities are used as prediction variables and the number of features is reduced through correlation coefficient analysis. These variables are then entered into the machine learning to determine an appropriate prediction model, and become reusable core assets.

In order to confirm the correlation between learning activity features of the course, those of the 16 features in Table 1 which have dependency with the final scores are deleted first, including unit_score, final_score, final_result and total_score. The remaining 12 features are called feature set A for learning, and Pearson Correlation Coefficient Analysis is carried out to obtain the correlation degree between two features. Then the correlation matrix is used to display the correlation between any two variables in the multivariate data. At last, the rcorr function is performed with the related variable data to calculate the correlation coefficient matrix and the corresponding *p*-value matrix of the data of any two variables.

After performing Pearson Correlation Coefficient Analysis for learning activity features, this study found that there was a high correlation between several groups of features higher than 0.9, as follows:

(1) unit_num, exam_num and prom_num.
(2) video_num and sess_num.
(3) sess_num and load_num.
(4) exam_num and prom_num.
(5) prom_num and all_attempts.

This study then selected unit_num, video_num, sess_num, exam_num and all_attempts through the Findcorrelation function in Caret of R. Since these five features were highly correlated with other features, they were removed from the feature set to avoid interference with similar features. Then, the dimensions of the feature set were reduced from 12 to 7, and were labeled feature set B for learning. Next, the machine learning

models of the feature sets A and B were established as the core assets of the prediction model. The course data set contained the data of 590 learners, and 70% of the 590 data set (413) were used as training data for model building, and the remaining 30% (177) as verification data.

First, the library (ISLR) suite was loaded in *R* language for the use of the KNN method, namely the knn() function is used. Of the 532,579 learning activity records with the feature set A, 70% of them were used as a training data set, and 30% were used for verification. The real classification factor of the training set was passing (1) or failing (0) the course, and the *k* value (number of close neighbors) was the square root of the total number of data. Finally, the model accuracy obtained was 0.847458.

As KNN's accuracy was not as high as expected, the SVM method was used next. The library (e1071) suite was loaded in *R* language and the svm() function was used to train the SVM classification model with 70% of the learning activity records. The predict() function was used for verification with the remaining 30% of learning activity records. The obtained accuracy was 0.920904.

To obtain a better result, the ANN method was used by loading the library (nnet) suite in *R* language. With the ann() function, the same data distribution of 70% and 30% as before was used. Several experiments were conducted to find the best parameter settings. For example, the parameter of proportion attenuation was 0.001 and the maximum repeated times was 1000. The number of hidden layers was then set from 1 to 10, and ten models were built for each using different seed values. The accuracy value of each model was the average of its ten verification results. This study found that the best accuracy of 0.949153 was achieved in the experiment with one hidden layer. Therefore, the best core assets obtained with the KNN, SVM and ANN methods in the model building and prediction for the feature set A are ANN with one hidden layer. The result is shown in Table 4.

**Table 4. Accuracy of ANN, KNN and SVM.**

| Model | Size | Feature set A |
|---|---|---|
| KNN | | 0.8474576 |
| SVM | | 0.920904 |
| ANN | 1 | 0.949153 |
| | 2 | 0.909605 |
| | 3 | 0.932203 |
| | 4 | 0.949153 |
| | 5 | 0.943503 |
| | 6 | 0.898305 |
| | 7 | 0.915254 |
| | 8 | 0.926554 |
| | 9 | 0.870056 |
| | 10 | 0.881356 |

The same set of 532,579 learning activity records were applied to feature set B. Since the ANN method core asset can be reused, the model building process was sped up by adopting the parameter settings from that of feature set A. The best accuracy achieved using ANN with one hidden layer for feature set B was 0.9096045.

### 4.2 Product Development

This study used the core assets built for reuse with SPL application engineering to predict the list of students requiring tutoring in the course each week. With the ANN model core asset from the previous section, these students were identified using their weekly learning activity records to predict if they would "fail (0)" the course.

In order to obtain the list of students requiring tutoring in advance, the activity records of students were collected in weekly intervals. In other words, the learning data of students were divided into how many learning activities were completed in the first week, how many learning activities were completed in the second week, and so on. These data were cumulative, and the data for the second week contained data for the first two weeks. This study used week-to-week student data to establish the accuracy of the ANN prediction model.

Fig. 6 shows the results of the ANN prediction model using the core assets of feature sets A and B to make the week-by-week prediction in order to provide a list of students requiring tutoring for the corresponding weeks. The accuracy was only 36.6% with the data for the first week for feature set B, and increased slightly to 44.7% for the second week. The prediction accuracy for the third week reached 73.5%, and the accuracy for the following two weeks rose gradually, reaching 92.3% in the fifth week. Therefore, the prediction accuracy of feature set B is better than that of feature set A. Finally, the number of students needing tutoring in the first week was 212, 256 in the second week, 267 in the third week, 280 in the fourth week, and 285 in the fifth week.

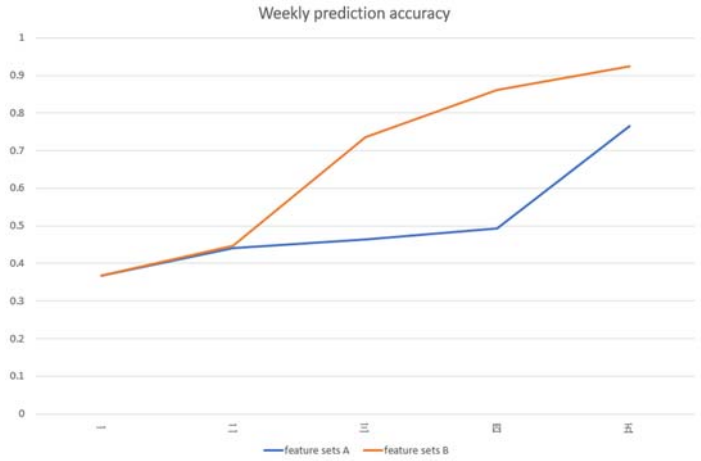


Fig. 6. Weekly prediction accuracy.

## 5. CONCLUSION

This study proposed an SPL-based Learning Analytics Framework for application in MOOC learning analysis and application development. Domain Engineering was first used to build the core assets and related general components to provide users with essential functions; then Application Engineering was used to establish applications for users'

specific needs, and feed back to the management of the core assets. A MOOC learning analytics service can be based on such a framework.

This study used the learning data of a basic course from the OpenEdu platform to obtain 16 features related to the learning activities through the development of core assets. Then features related to the learning performance were deleted to form the feature set A with 12 features. Next, Pearson Correlation Coefficient Analysis was used to obtain the correlation degree between two features. The features were selected by deleting highly correlated ones to obtain the feature set B with seven features. Then, these feature sets were used to organize related learning data to train various prediction models.

This research used KNN, SVM, and ANN to build models for predicting whether students would pass their courses. The experiment results show that ANN has the best prediction accuracy of 0.949153, and the built models become the core assets. In addition, data collection, data cleaning, and feature selection modules are saved as core assets.

The advantages of the proposed SPL-based method were verified by applying reusable core assets of prediction models to provide a weekly tutorship list, allowing teachers to monitor learning progress. A total of 212 students required tutoring in the first week, 256 students in the second week, 267 students in the third week, 280 students in the fourth week and 285 students in the fifth week.

Therefore, the proposed MOOC Learning Analytics Framework provides the development environment of SPL, and gives full functionality to reuse, resulting in good experiment results. The prediction accuracy of the system is as high as 94%. In addition, the core assets were reused with new requirement specifications to rapidly develop an application for developing a midterm tutoring list to improve the final pass rate and reduce the dropout rate.

Although open online learning platforms are diverse, including a variety of different xAPI technologies and multimedia presentation modes, the proposed MOOC learning analytics framework can be used in future to reuse the core assets based on similar data records, and to more efficiently develop new products. Furthermore, it is beneficial to classify commonality and variability of framework components, which can become a part of the core assets, saving development time and cost. Future work will include the analysis of different types of courses, and those on the other MOOC platforms. Since the characteristics of course content and teaching objectives can be very different, it is necessary to build more core assets and set up the environment for other MOOC platforms.

## ACKNOWLEDGEMENT

This research is partially supported by the Ministry of Education, Taiwan. Thanks to Prof. Don-Lin Yang and the OpenEdu team who develop the system and provide the open data. This research is partially supported by the Ministry of Science and Technology, Taiwan, under Grant No. MOST 109-2511-H-035-001-MY2 and the Ministry of Education, Taiwan, under Grant No. PED1090599.

## REFERENCES

1. M. Khalil and M. Ebner, “Clustering patterns of engagement in massive open online

courses (MOOCs): the use of learning analytics to reveal student categories," *Journal of Computing in Higher Education*, Vol. 29, 2016, pp. 114-132.
2. J. Liang, C. Li, and L. Zheng, "Machine learning application in MOOCs: Dropout prediction," in *Proceedings of the 11th International Conference on Computer Science and Education*, 2016, pp. 52-57.
3. J. Kay, P. Reimann, E. Diebold, and B. Kummerfeld, "MOOCs: So many learners, so much potential," *IEEE Intelligent Systems*, Vol. 28, 2013, pp. 70-77.
4. Software Engineering – Software Process and Software Process Models (Part 2), http://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc.
5. C. Severance, "Teaching the World: Daphne Koller and Coursera," *Computer*, Vol. 45, 2012, pp. 8-9.
6. 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, 2013, pp. 13-25.
7. W. Greller and H. Drachsler, "Translating learning into numbers: A generic framework for learning analytics," *Educational Technology & Society*, Vol. 15, 2012, pp. 42-57.
8. A. G. Picciano, "The evolution of big data and learning analytics in American higher education," *Journal of Asynchronous Learning Networks*, Vol. 16, 2012, pp. 9-20.
9. G. Siemens and P. Long, "Penetrating the fog: Analytics in learning and education," *EDUCAUSE Review*, Vol. 46, 2011, p. 30.
10. ISO/IEC TR 20748-1, "Information technology – learning, education, and training – learning analytics interoperability – Part 1: Reference model," 2016.
11. M. A. Chatti, "The LaaN theory," *Personalization in Technology Enhanced Learning: A Social Software Perspective*, Shaker Verlag, Germany, 2013, pp. 19-42.
12. R. N. Laveti, S. Kuppili, J. Ch, Supriya N. Pal, and N. S. C. Babu, "Implementation of learning analytics framework for MOOCs using state-of-the-art in-memory computing," in *Proceedings of the 5th National Conference on E-Learning & E-Learning Technologie*, 2017, No. 8074997.
13. N. H. Bakar, Z. M. Kasiruna, and N. Salleh, "Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review," *Journal of Systems and Software*, Vol. 61, 2015, pp. 33-51.
14. P. Clement and L. Northrop, "A framework for software product line practice, version 5.0," Software Engineering Institute, Carnegie Mellon University, 2012.
15. D. Batory, R. Cardone, and Y. Smaragdakis, "Object-oriented frameworks and product lines," in *Proceedings of the 1st Conference on Software Product Lines*, 2000, pp. 227-247.
16. A. Shatnawi, A.-D. Seriai, and H. Sahraoui, "Recovering software product line architecture of a family of object-oriented product variants," *Journal of Systems and Software*, Vol. 131, 2017, pp. 325-346.
17. A. Kashyap and A. Nayak, "Different machine learning models to predict dropouts in MOOCs," in *Proceedings of International Conference on Advances in Computing*, 2018, pp. 80-85.
18. R. Al-Shabandar, A. Hussain, A. Laws, R. Keight, J. Lunn, and N. Radi, "Machine learning approaches to predict learning outcomes in massive open online courses," in

*Proceedings of International Joint Conference on Neural Networks*, 2017, pp. 713-720.
19. A. Eskandarinia, H. Nazarpour, M. Teimouri, and M. Z. Ahmadi, "Comparison of neural network and K-nearest neighbor methods in daily flow forecasting," *Journal of Applied Sciences*, Vol. 10, 2010, pp. 1006-1010.
20. S. Fauvel and H. Yu, "A survey on artificial intelligence and data mining for MOOCs," arXiv preprint arXiv:1601.06862, 2016.
21. Chinese Open Education Consortium, https://copeneduc.org/, 2017.
22. M.-C. Liu, C.-H. Yu, J. Wu, A.-C. Liu, and H.-M. Chen, "Applying learning analytics to deconstruct user engagement by using log data of MOOCs," *Journal of Information Science and Engineering*, Vol. 34, 2018, pp. 1175-1186.
23. D. Benavides, S. Segura, and A. Ruiz, "Automated analysis of feature models 20 years later: a literature review," *Information Systems*, Vol. 35, 2010, pp. 615-636.
24. D. Beuche and M. Dalgarno, "Software product line engineering with feature models," *Overload Journal*, Vol. 78, 2007, pp. 5-8.
25. P. Clements and L. Northrop, S*oftware Product Lines – Practices and Patterns*, Addison Wesley, Boston, MA, 2002.
26. C.-H. Yu, J. Wu, and A.-C. Liu, "Predicting learning outcomes with MOOCs clickstreams," in *Proceedings of the 2nd Eurasian Conference on Educational Innovation*, 2019, pp. 305-308.
27. C.-H. Yu, J. Wu, D.-L. Yang, M.-C. Liu, and A.-C. Liu, "Video watching behavior pattern comparison of MOOCs clickstream," *Taiwan E-Learning Forum*, 2018, National Taichung University of Education.

**Chen-Hsiang Yu (余禎祥)** is the Ph.D. candidate at the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include network management and data analysis.

**Jungpin Wu (吳榮彬)** is an Associate Professor at the Department of Statistics, Feng Chia University, Taiwan. His research interests include sampling design, spatial statistics, experimental design, questionnaire design, and empirical process approach.

**Ming-Chi Liu (劉明機)** is an Assistant Professor at the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include education technology, affecting learning, human/brain computer interface, nature language processing, and semantic analysis.

**An-Chi Liu (劉安之)** is the Chair Professor at the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include network management, distributed system, and database system.