逢　甲　大　學
資　訊　工　程　學　系

博　士　論　文

磨課師學習分析的軟體框架開發

Software Framework Development for MOOCs Learning Analytics

指導教授：劉安之

研 究 生：余禎祥

中　華　民　國　一　百　零　八　年　六　月

Software Framework Development for MOOCs Learning Analytics

by

Chen-Hsiang Yu

A dissertation submitted to the graduate division in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

at
Department of Information Engineering and Computer Science
Feng Chia University
Taichung, Taiwan, R.O.C.
June 28, 2019

Approved by

Mien Lin Hsueh

Chi Chi Lin

Jim-Min Lin

Jungpin Wu

Yu-Mei Wang

Dinda Yang

Chao-Hsi Huang

Thesis Advisor: Chi Chi Lin

Chairman: Mien Lin Hsueh

逢甲大學
資訊工程學系博士學位論文

磨課師學習分析的軟體框架開發

Software Framework Development for MOOCs Learning Analytics

研究生：余禎祥

經博士學位考試合格特此證明

評審委員

劉士之

指導教授

劉士之

主　任

考試日期：中華民國108年6月28日

Acknowledgements

在博士班的求學過程中，首先要感謝我的指導老師劉安之教授，在我博士生涯中他除了在課業上的細心指導，亦在待人處事上多方指點，以及有機會面對與解決各種的新事物，另外在問題的發掘、思考邏輯和發問技巧的訓練上更是獲益良多，使得我的研究能有豐富的成果，並且順利完成本論文；此外還要感謝我的口試委員：楊東麟、林志敏、薛念林、吳榮彬、黃朝曦與王履梅老師，還有何苑荔小姐與所有教導過我的所有師長們，讓我修業過程中獲得到更多方面的知識，繼而使我在論文研究上的思考能夠更多元化。

另外，我還要感謝逢甲大學資訊工程研究所提供我一個優良的學習環境以及完善的實驗設備，使得我能夠有優良的學習成果。以及感謝新一代數位學習計畫磨課師分項計畫辦公室團隊，讓我有這樣研究環境，完成研究成果。衷心感謝在我博士研究期間楊東麟、吳榮彬與劉明機老師的督促與討論。並感謝實驗室各位學弟們義州與瓏騰，在實驗室裡的生活有了他們的幫忙以及扶持，使得我在這幾年能夠過得充實並且多采多姿。此外，要感謝所上助教們的幫忙和提供的設備資源，得以順利完成博士學位。

最後，我要感謝我親愛的家人，我的父母親從小到大的栽培，以及妻子朱羽莛，由於他們的支持、鼓勵、教誨以及無限的愛，才能讓我今天得以順利完成本論文而取得博士學位，僅以本論文獻給我最親愛的家人。

余禎祥謹誌於

逢甲大學資訊工程研究所

中華民國 108 年 6 月

摘要

使用學習分析(Learning Analytics)處理 MOOCs 平台上不同課程目標的科目、多元學習活動與學生學習差異，常常增加資料分析的複雜度，造成進行決策支援的過程很漫長，無法達到教師要求的即時性。尤其是學校裡面的 MOOC 課程要讓教師掌握大量學生的學習狀況，須要能夠及時進行預警與輔導作業，才比較有機會提高學生的學習參與度和課程通過率。我們認為使用好的軟體框架，可以快速建立具便利性與彈性的各種分析模型，這是我們的研究之主要目標。

本文中我們針對學習者的影片觀覽資料進行分析，並提出有效的軟體框架來解決上述的問題。我們的研究分為三大部分：第一項是前置作業的探討，我們先透過 MOOCs 學習事件驗證學生的學習成效與學習的行為、認知與情感三種參與度重要的關聯。第二項為進行軟體工程技術中的軟體框架開發(Software Framework Development)，並以軟體產品線(Software Product Lines, SPL)的概念應用於學習分析的框架中，此產品線式的資料分析框架，可以引導使用者，讓資料分析過程以如同軟體產品開發一樣，具有可重複使用性(reuse)，而且在特定的領域下建立核心資產並加管理。當要進行新服務開發時，即可善用核心資產來整合新的需求所開發的元件，得到最好的整體效益。第三項為以前述的軟體框架下，利用學生的影片點擊流記錄，建立學習者的七種認知參與模型。並使用 K-最近鄰(KNN)、支持向量機(SVM)和人工神經網絡(ANN)演算法來構建實用的機器學習模型，透過他們的學習行為資料來預測學生的學習成果。

本研究主要貢獻包括：(1)設計 MOOCs 的學習分析之軟體開發框架、學習分析原型，(2)分析 MOOC 平台上學生學習行為記錄的相關變量(例如觀覽影片的事件、自我評估測驗)，(3)建立課程學習影片觀看序列模型，(4)以課程章節的學習測驗為單位，建構學習特徵項目與預測模型，(5)並以 OpenEdx 平台環境下展示三個學習分析模組的實例，分別有：(i)參與度的系統日誌分析成果，(ii)以影片點擊資料建立的預測模式預測學習成果，(iii)以影片點擊序列模型預測學習成

果。

關鍵詞: 磨課師、學習資料分析、影片點擊流、軟體產品線、機器學習。

Abstract

The use of learning analytics to deal with the different curriculum objectives of the MOOCs platform, multi-learning activities and student learning differences often increases the complexity of data analysis, resulting in a long process of decision support, unable to meet the requirements of teachers in a timely manner. In particular, the MOOCs course in the school requires teachers to master the learning situation of a large number of students, and it is necessary to be able to conduct early warning and counseling in order to improve the students' participation in learning and the passing rate of the course. We believe that using a good software framework can quickly establish a variety of analytical models with convenience and flexibility and that is the main goal of our research.

In this thesis, we analyze the learner's video viewing data and propose an effective software framework to solve the above problems. Our research is divided into three parts: The first one is the study of our preliminary research. We first use the MOOCs learning event to verify the important relationship between the students' learning outcomes and the learning behavior, cognition and emotion. The second item is the software framework development in software engineering technology, and the software product line (SPL) concept is applied to construct the framework of MOOCs learning analytics. This SPL-based framework can guide users to take advantage of software development reuse and build core assets with effective management in specific areas. When new product development is required, core assets can be leveraged to integrate the components developed by the new requirements to achieve the best overall benefits. The third item is to use the student's video clickstream records to form seven cognitive participation models for learners under the aforementioned software framework. K-nearest neighbor (KNN), support vector

machine (SVM) and artificial neural network (ANN) algorithms are used to construct practical machine learning models to predict student learning outcomes through their learning behavior data.

The main contributions of this research include: (1) Designing the software development framework for learning analysis of MOOCs and learning analytic prototypes. (2) Analyzing the student learning behavior records from related variables of the MOOC platform (e.g., interaction in viewing the video, self-assessment test). (3) Establishing a video viewing sequence model of MOOCs. (4) Constructing a feature model of learning analytics and a predictive model of learning performance based on the course unit and associated assessment tests. (5) The three examples of learning analysis models are demonstrated in the OpenEdx environment, which are: (i) the results of system log analysis of course participation. (ii) the prediction of learning outcome using the video clickstream model. (iii) the learning outcome prediction using the sequence model of the video clickstreams.

Keywords: MOOCs, Learning Analytics, Video Clickstream, Software Product Lines, Machine Learning.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

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) in education today, and the Ministry of Education in Taiwan has promoted MOOC programs for universities since 2014 [1]. 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.

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.

1.2 Motivation

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 [2]. Therefore, this study focused on the Software Product Lines (SPL) 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 [3].

1.3 Objective

This research 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.

1.4 Thesis Organization

This thesis is divided into six chapters, which are explained as follows:

The second chapter describes the related research, which will explain the importance and data analysis of the MOOCs, the consideration of learning analysis, the importance of software development mode and software product line on software engineering, and the type of machine learning method.

The third chapter contains the research process. We briefly describe the three phases of the study and explain the relationship among them.

The fourth chapter describes the implementation of the three phases. First, we apply learning analytics to deconstruct user engagement. Second, we implement the SPL-Based MOOCs Learning Analytics Framework. The third is to predict learning outcomes with MOOCs clickstreams.

The fifth chapter describes the experiments of the three phases. The first experiment deconstructs user engagement to find learning behaviors and establish the feature set of the predictive model of the video browsing click events. The second experiment is to show that our SPL-Based MOOCs Learning Analytics Framework is feasible and practical. The third one is to verify the resultant predictive model can be used to predicting learning outcomes with MOOCs clickstreams.

The sixth chapter summarizes all aspects of this thesis, including conclusions and future research directions.

Chapter 2 Related Work

2.1 Massive Open Online Course

Currently, the most popular MOOC platforms in the world include OpenEdX jointly established by Massachusetts Institute of Technology, Harvard University and UC Berkeley [2]; Coursera founded by two professors in Information Engineering from Stanford University [4]; the Khan Academy founded by Salman Khan, a graduate of Massachusetts Institute of Technology and Harvard University and Udacity funded by Sebastian Thrun, David Stavens and Mike Sokolsky [5]. 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.

The MOOCs consists of five elements: Instructors, Learners, Topics, Materials, and Context [6]. Instructors: Simplify the learning process by producing appropriate textbooks, trigger communication between learners and manage assessments of expected learning outcomes. Learners: Anyone who wants to learn about a topic is authorized to register, and the learner can pursue a formal degree or credit from some courses, or just access specific content. Topic: Themes that are triggered by learners, teachers, textbooks, and contexts are introduced through the system, limited but broad enough to cover a wide variety of fields. Materials: exist on different websites and come in a variety of styles, accessed through a variety of technical solutions. Context: Representing the different members of a curriculum environment, combined with online social networking, common sources of information, different types of information delivery methods, communication systems, expected learning outcomes, and group-building courses.

Some students are easily distracted in traditional classrooms, which leads to a lot of time spent on review and homework after returning home. MOOCs are different from traditionally taught courses in that students can play back content if they do not understand the course. MOOCs provide online peer assistance for learners and opportunities to interact online with course teachers [7]. Compared with the previous form of online education, MOOCs are closer to personalized learning—there is no teacher supervision, no entry threshold, and no need to pay expensive fees. MOOCs facilitate self-regulated and individualized learning in order to enable learners to achieve better learning results. Many studies are now focusing on analyzing the learning history records left by users of MOOCs [8] in order to predict students' possible achievements through analytical methods [9] and to provide early guidance to students who need help.

MOOC courses are mainly based on video viewing and quizzes, which take the majority of the learners' time. Many problems have been gradually found. First, many students neither continue to participate in learning after enrolling in a course nor meet the standards for passing the course after the course ends. This behavior of students not completing the courses [10] prompts the question of how to stimulate the completion rate, which is a problem that every MOOC platform wants to solve [11]. One reason for the low completion rates may be the students' own problems, and some students may need more proper supervision [12]. It may also be a problem with the video material, which may need to be adjusted or supplemented. There is no clear answer at present, thus, stimulating the completion rate is a major challenge for MOOCs [13, 14].

The nature of this type of course is different. The style of the course videos depends on the teachers' preference and the feasibility, or on the institutional guidelines other than structured theory [15]. Moreover, there are significant

characteristics that can be adapted, with respect to learners passing or failing a course, if the course videos are properly classified by their nature [16].

As MOOC is a kind of personalized autonomous learning, in order to make learners get a better learning result, now many researchers focus on the analysis on the learning process records of users in MOOCs [17, 18, 19]. 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 on the learning experience of learners in the field of digital learning.

The experimental environment of this study adapts the OpenEdu platform, as established by the Chinese Open Education Consortium and based on edX open source software [20]. The platform aims to continuously promote open courses, increase the level of influence of teaching innovation, follow the development trend of international digital learning, and narrow the gap between urban and rural areas, thus ensuring equal rights to education. To this end, the Chinese Open Education Consortium has joined many schools and institutions interested in developing MOOCs—including the organization's fundraising and human operations—by providing construction guidance, teaching platform maintenance, promotion, and other services through the construction of the alliance system.

In their discussions of the low completion rate of the MOOCs course, the researchers analyzed the learners' video viewing, scores, and forum behavior records [21, 22]. In Anderson [23], the students' activity behavior patterns were divided into five types: Viewers, Solvers, All-rounders, Collectors, and Bystanders. In Rebecca [24], the students' activity behavior patterns were divided into seven types: Samplers, Strong Starters, Returners, Mid-way Dropouts, Nearly There, Late Completers, and Keen Completers. In Khalil [25], the students' activity behavior patterns were divided into four types: Dropout, Perfect Students, Gaming the System, and Social. In Sinha, a

cognitive video watching model was applied to explain the dynamic process of cognition involved in MOOC video clickstream interaction [26]. The students' activity behavior patterns were divided into seven types: Rewatch, Skipping, Fast Watching, Slow Watching, Clear Concept, Checkback Reference, and Playrate Transition. The purpose of the above discussion is to improve students' participation in learning, to help solve the problem of the low course completion rate.

2.2 Learning Analysis

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.

Learning analysis is usually an iterative periodic process with three parts [27]: data collection and pre-processing, analysis and action, and post-process. Data collection and pre-processing: Data is the basis for learning analysis and a primary and important step in collecting data from diverse learning environments or systems. Analysis and Action: Based on the data generated by pre-processing, different learning analysis techniques are applied in order to explore the information hidden in the data that can help to learn effectively. Post-processing: In order to continually improve the analysis process and actions, additional data may need to be collected and aggregated into new data, new indicators needed for the next iteration, modified analytical variables, or new analytical methods selected.

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) [28, 29, 30].

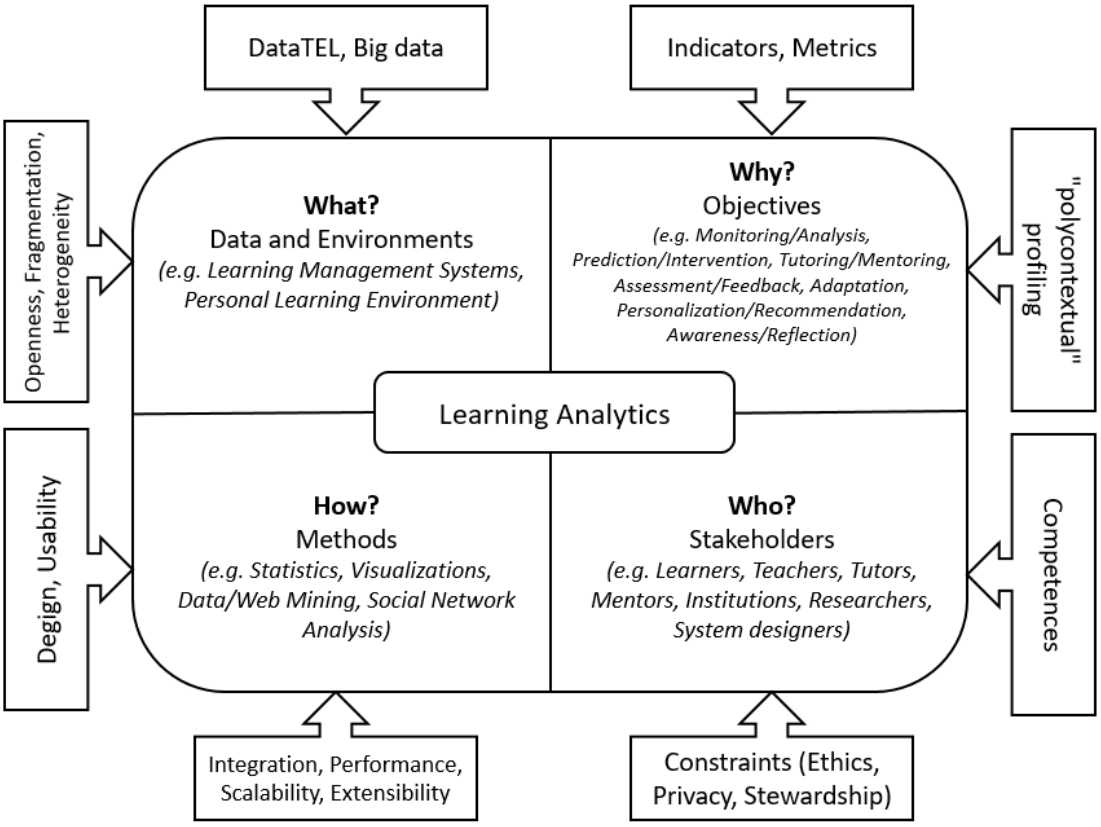

Fig 2.1 Learning Analytics Reference Model.

What: The source of learning and analysis data, mainly from Centralized educational systems and distributed learning environments, the information of centralized education systems mainly comes from the Learning Analysis System. LMS has accumulated a wealth of data on learner activity data and interaction data, such as reading, writing, accessing and uploading learning materials, testing, and sometimes simple built-in reporting tools.

Who: The application direction of learning analysis varies from user to user, including learners, teachers, mentors, administrators of educational institutions, or decision makers. Users of different ranks have different interpretation requirements for the analysis. The learners may be interested in how the analysis may improve their performance or help them establish a personal learning environment; teachers may

analyze how to improve the effectiveness of their teaching practices or support them. Teaching that adapts to the needs of learners may be of interest.

Why: Different users provide information for learning analysis, and learn to provide as many goals as possible: Monitoring and Analysis, Prediction and Intervention, Tutoring and Mentoring, Assessment and Feedback, Adaptation, Personalization and Recommendation, Awareness and Reflection.

How: Learning analysis is the application of different methods and techniques to detect meaningful elements hidden in the learning trajectory. In recent years, the four techniques that are the most widely used and discussed most are statistical, information visualization, and data. Exploration and Social Network Analysis.

To evaluate users' learning behavior and achievements, we can analyze their video watching activities and test results. Most learning platforms monitor and record the whole learning process in the various logs. The results of learning analytics can provide users with learning status and performance level, making them aware of their problems or insufficiency to improve. On the other hand, teachers can benefit from the analysis results to see if the learning outcomes are as expected or modification on teaching activities and course materials are required. The interaction data between users as well as between users and teachers are valuable resources for learning analytics to understand and provide better communications among the platform stakeholders.

2.3 Software Development Model

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. The common software development models include Waterfall,

Agile, Object-oriented, Software Product Lines [31, 32], etc. The waterfall model divides the life cycle of software into six essential activities of planning, requirement analysis, design, programming, software testing, and operation maintenance, which has a fixed order from top to down just like a waterfall and lacks flexibility (Fig 2.2). Although Agile is relatively flexible, which manages the development of products more effectively through incremental and iterative processes; it is no better than waterfall in terms of reuse (Fig 2.3). Object-oriented programming is a programming method with the concept of object [33]. 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. Object-oriented is suitable for reuse of objects and encodings.

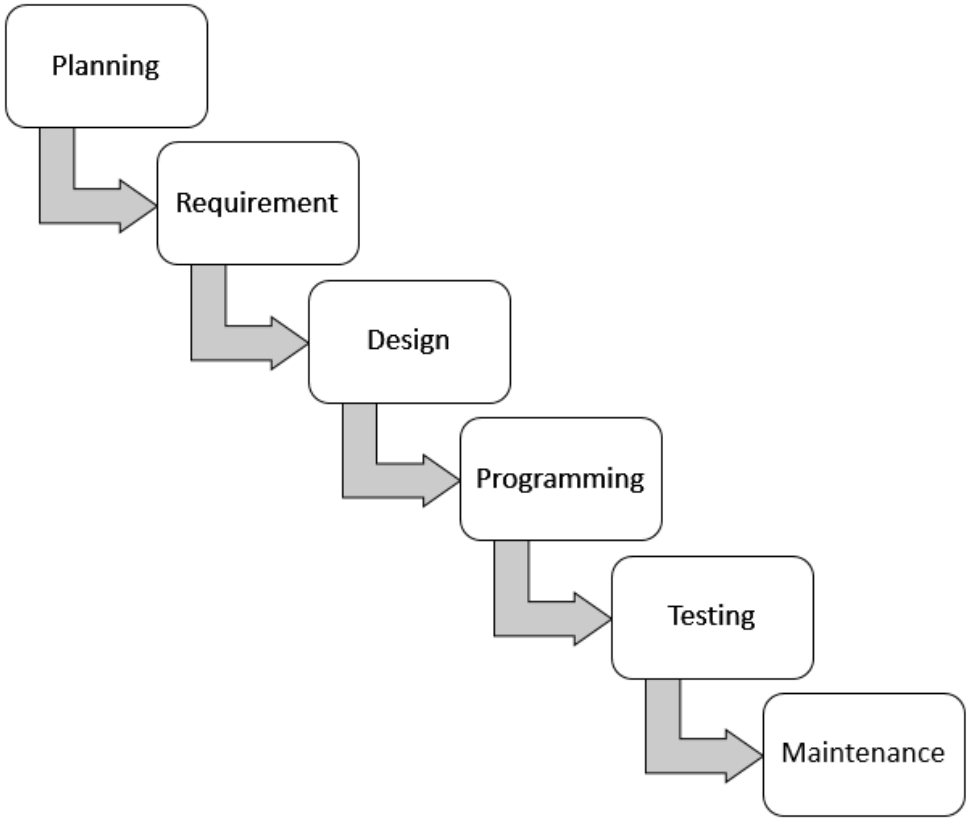

Fig 2.2 Waterfall software development process.

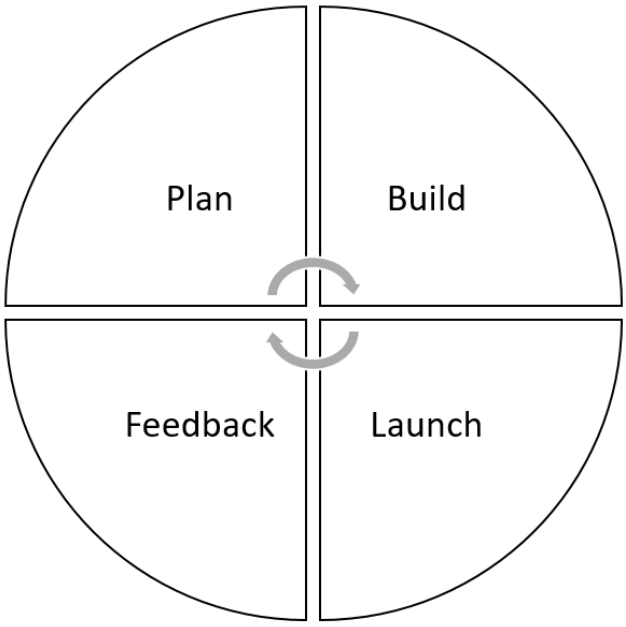

Fig 2.3 Agile software development process.

Fig 2.4 Software product line development process.

The software product line development process (Fig 2.4) refers to establish the core assets and then develop similar software systems of that with high properties in terms of the specific fields. The core of SPL is strategic reuse, which can reuse various types of software components in different software development stages, thereby improving the reuse rate of software components [34]. Compared to object-oriented programming, SPL is suitable for reuse and more flexible of the overall software development process. Its primary process mainly consists of two major steps. The first step is called domain engineering, when core assets which can meet general demands are developed. The second step is called application

engineering, when the core assets are reused to develop products that meet the customer's specific requirements. Therefore, the software product line approach is based on the practice of reusing the existing software assets as far as possible and then develops a series of similar products that meet the requirements of different users. What's more, 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 by new requirements for the best overall benefit.

2.4 Engagement

User engagement is conceptualized as a need-based psychological state of users toward a system (how motivated they are) [35]. The state of user engagement can be observed by the behavior of user involvement and participation. Research has shown that user engagement has a positive effect on system success [35]. O'Brien and Toms argued that software requirement analysis should move beyond usability to understand and design for more engaging user experiences [36]. It is thus important to measure the engagement of users during the development process of a system.

Previous studies [37, 38] have shown that system log analysis might provide a way to capture user engagement over time. For example, Ramesh, Goldwasser, Huang, III and Getoor [39] measured the counts of posting and viewing to predict student engagement on a MOOC. Such work focused on the measurement of user engagement as behavioral participation (e.g., The frequency of completing the tasks). However, the conceptualization of engagement should be defined as more than a sum of the individual behavioral component.

As noted by Fredricks, Blumenfeld and Paris [40], engagement is characterized as a multi-dimensional construct, referring to behavioral engagement, cognitive engagement, and emotional engagement. Measuring engagement solely as the

frequency of task participation may focus only on behavioral engagement and ignore the multifaceted nature of engagement [41].

Nevertheless, the challenge of using system logs to fully understand the user engagement lies in exploring the relationships between event logs and components of engagement. In this research the system logs collected from the MOOCs were used as the case study. We measured engagement through mapping event logs with three components of engagement. Further, we also analyzed the helpfulness of engagement measurement in predicting grades. This thesis aims to stimulate a discussion on ways that the system log analysis can be used to better understand user engagement for the purpose of system design.

2.5 N-gram

N-gram is easy to access with a rapid calculation and without any complicated algorithm [42]. Therefore, such a method is used in natural language processing to increase calculation efficiency. Articles or sentences are segmented into many small parts when applying the N-gram method, and only a few parts will be affected in the case of an error in an article or a sentence. As such, the N-gram method provides a good error-tolerant rate in natural language processing, can be applied in correcting wrongly written or misspelled characters, and is often applied in calculating the similarity between different articles and sentences or retrieval of texts. Articles or sentences are segmented into many small parts, such that many text combinations of different lengths are also produced if a corpus with small data volume is used. Identical sentences in an article can be segmented into text combinations of different lengths to achieve the effect of multi-segmentation and to obtain more text combinations [43]. In addition, N-gram extraction methods consist of N-gram by character and N-gram by term.

2.6 Machine Learning

Machine learning is to classify collected data or train a prediction model through an algorithm, and when new data is obtained in the future, it can be predicted through the trained model. Machine learning techniques such as 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 [44]. Dropout prediction used machine learning of SVM, Logistics Regression, Random Forest, and Gradient Boosting Decision Tree to make dropout predictions [10]. Two types of neural network, Feedforward Neural Network (FFNN) and Self-Organised Map (SOM), were employed to predict if learners would receive certifications at the end of the course [45]. The machine learning data is composed of feature data and real categories in the process of model training. For example, the first algorithm of KNN (K-Nearest Neighbor) in this study is generally used to classify data, where K represents a constant, and KNN takes the K points of the nearest distance to determine to which category the object belongs [46]. The second algorithm of SVM is for supervised learning models, which is often used for pattern recognition, classification, and regression analysis [47]. The third one is an ANN (Artificial Neural Network) [48], composed of many neuron nodes, which can be divided into an input layer, an output layer, and a network model consisting of many hidden layers [49]. The output of the result can only be in the two states of yes or no, while the traditional artificial neural network can train the model by way of back-propagation, thereby obtaining a neural network model to effectively solve the problem (Fig 2.5).

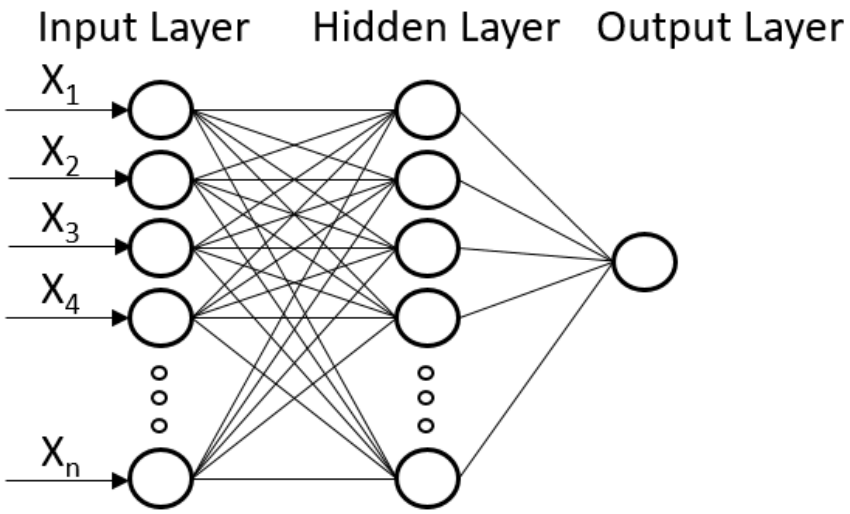

Fig 2.5 Artificial Neural Network.

Chapter 3 Research Process

In this research, the Learning Analytics is used to study the learners' learning behavior and their outcomes on the MOOCs platform. First, we use the clickstream records of video viewing to study the relationship between learning behavior and learning outcomes, and find out the feature sets that can be used to build predictive models. Then we develop the software framework for MOOCs learning analytics and establish the application prototype for learning outcome prediction. Thus, the prediction of MOOCs learning analysis allows teachers to monitor students' progress and help them pass the course, and making it easier to reuse software components during the model development. Finally, we use the developed software framework to implement the prediction model with experiments to verify that our approach is feasible.

3.1 Limitations

The limitations of this study are as follows:

(1) Since we only use the data of the OpenEdx platform for experiments, our research results are available on this platform. (2) If users want to apply our framework to other platforms, they need to adjust the data record content due to different recording formats. (3) The classification of the teaching videos of the course is currently conducted manually.

3.2 Process

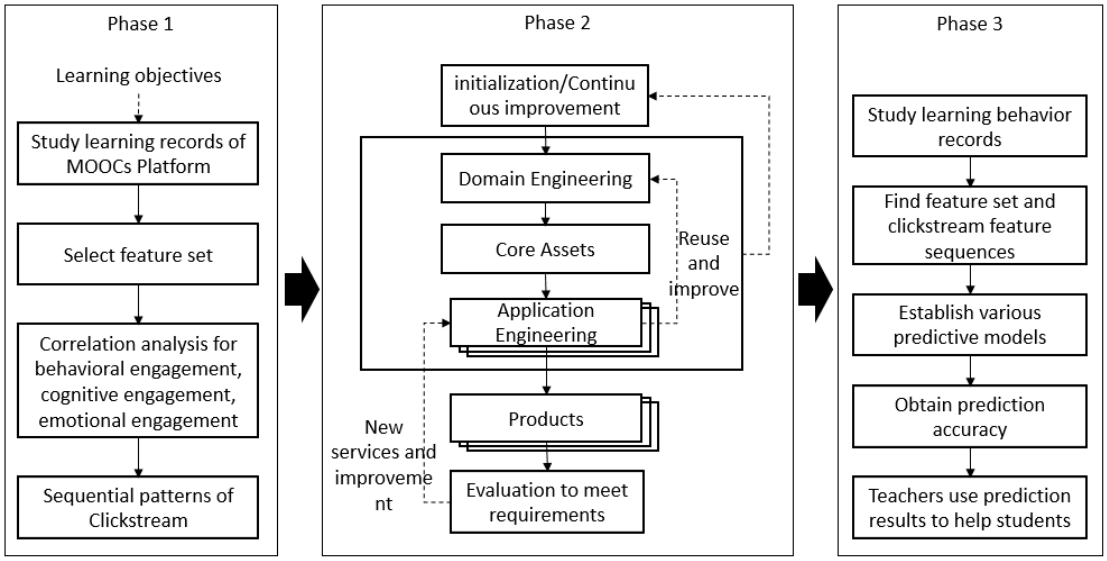

Fig 3.1 The research process followed in this thesis.

In our research process, we conducted a study on the learning analytics and the MOOCs platform, and set the software framework development for MOOC learning analysis as the research direction. Our approach has three phases as shown below:

(1) We investigated the MOOC platform learning record including the video viewing clickstreams and quiz results. Using the initial prototype of the learning engagement model, we analyzed the user engagement and video watching behavior from the clickstream records of the video to find out what features can be used to build the predictive model for learning outcome prediction (see Fig. 3.1 Phase 1).

(2) Based on the experience of establishing the feature set of the predictive model and the prototype, we started to find a suitable software development framework in order to efficiently generate proper learning outcome prediction models that can meet various requirements of teachers for a vast amount of MOOC courses. The use of software frameworks aim to facilitate software developments by allowing

developers to devote their time to meeting application software requirements rather than dealing with the low-level details of providing a working system, thereby reducing overall development time. We were looking for a software framework as a reusable software environment that provides a flexible way to build and deploy applications for MOOC learning analytics (see Fig. 3.1 Phase 2).

(3) The developed software framework was used to build various predictive models from video viewing clickstreams for predicting learning outcomes. In addition, the feature sequence of the viewing learning behavior was established by using the n-gram approach. The prediction of learning outcomes was presented through an analysis of learning records, course video clicking, and testing records. The result provided a reference for teachers to implement tutoring measures in a timely manner for students with poor learning outcomes and the course completion rate can be improved (see Fig. 3.1 Phase 3).

More detailed processes of our research are described as follows:

Using the learning log record of the MOOCs, the feature set of the predictive model of the video browsing click events is mapped to the behavior of learning, cognition and emotion Then the data of learning behavior is checked by using correlation and clustering analysis to establish a positive correlation with learning outcomes. The multiple linear regression and classification test are used to verify the relationship between learning behavior and learning outcomes. The accuracy was demonstrated by using three classification methods: SVM, Random Forest and ANN. The difference between the video viewing behavior of students with high learning performance and low test scores is further observed.

To make the development and analysis of MOOCs learning analytics more quickly and efficiently, we apply software development framework in software engineering technology for system implementation. Once a framework is established,

future projects can be faster and easier to complete. We investigated the concept of Software Product Lines (SPL) and decided to apply it in our environment. This product-line data analysis framework guides users in reuse through the process of data analysis as well as software product development, and builds and manages core assets in specific areas. As new service development is needed, core assets can be leveraged to integrate the components developed under the new requirements to obtain the best overall benefits. Domain Engineering is used to build the core assets and related general components with the essential functions. Application Engineering is employed to establish the application for users' specific needs.

We developed a prototype of this SPL-based framework, and used the feature set from the first phase to build the predictive models. A basic MOOC course was used in the experiment with the video clickstream record and test score record in the system log. Various data preprocesses are performed to filter and merge records into a course unit structure. For example, a unit may be set as a week based on the content of a video. Predictive models were generated using KNN, SVM and ANN for predicting whether the students pass courses. Students who may not be actively involved in the study will be provided with special attention from teachers and/or course assistants to help improve the course completion rate.

Under the SPL-based software framework developed in the second phase, we further applied the student's video clickstream record to establish the sequence behavior events of the video viewing, where the seven cognitive participation models of the learners were generated. According to the course content and teaching objectives, the video is divided into three categories or types for performance improvement. Using K-nearest neighbor (KNN), support vector machine (SVM) and artificial neural network (ANN) algorithms to construct practical machine learning models, we can predict student learning outcomes through their learning behavior data.

To demonstrate the flexibility and reusability of our framework, we collected video clickstream data from one additional course and used the data of three semesters. Here the data of the first two semesters was used for training and the data of the last semester for verifying the prediction accuracy.

Chapter 4 Implementation

In this chapter, we present the implementation of the three phases in accordance with the research process in Chapter 3. First, the implementation of Phase 1 is described in Section 4.1 by applying learning analytics to deconstruct user engagement. Then, the construction of a SPL-Based MOOCs Learning Analytics Framework of Phase 2 is reported in Section 4.2. Finally, the implementation of Phase 3 to predict learning outcomes with MOOCs clickstreams is described in Section 4.3.

4.1 Applying learning analytics to deconstruct user engagement

As defined by Trowler [50], "Student engagement is concerned with the interaction between the time, effort and other relevant resources invested by both students and their institutions intended to optimize the student experience and enhance the learning outcomes and development of students and the performance, and reputation of the institution."

As argued by Sinclair and Kalvala [51], log analysis about engagement in MOOCs overwhelmingly refers to student actions such as videos watched, quizzes answered and posts made on the forums. For example, Anderson, Huttenlocher, Kleinberg and Leskovec [23] selected six Coursera courses, including three machine learning courses and three probabilistic graphical models courses and analyzed the student learning behavior during the courses. Their findings showed that the pattern of student learning behaviors could be clustered into five groups: viewers, solvers, all-rounders, collectors, and bystanders. Moreover, Ferguson and Clow [52] selected four Coursera courses, including physical sciences, life sciences, arts and business and analyzed student learning behavior during the courses. They classified students into seven classification groups: samplers, strong starters, returners, mid-way dropouts,

nearly there, late completers, and keen completers. Finally, Khalil and Ebner [53] take social aspects of information technology as the target course and classified students into four types of groups: dropout, perfect students, gaming the system and social, and learning participation. The aforementioned research focused on K-means clustering analysis, and did not consider student engagement. These measures represent the level of engagement on a single count variable, but do not reflect whether that the collected data can be interpreted as a benchmark for learning improvement.

This study uses the same concept to measure engagement. The justification for classifying different video log events to three components of engagement was based on Fredricks, Blumenfeld's theory [40] and Li and Baker's study [41]. Therefore, this section describes the video interaction events to identify the components of engagement (see Fig. 4.1).

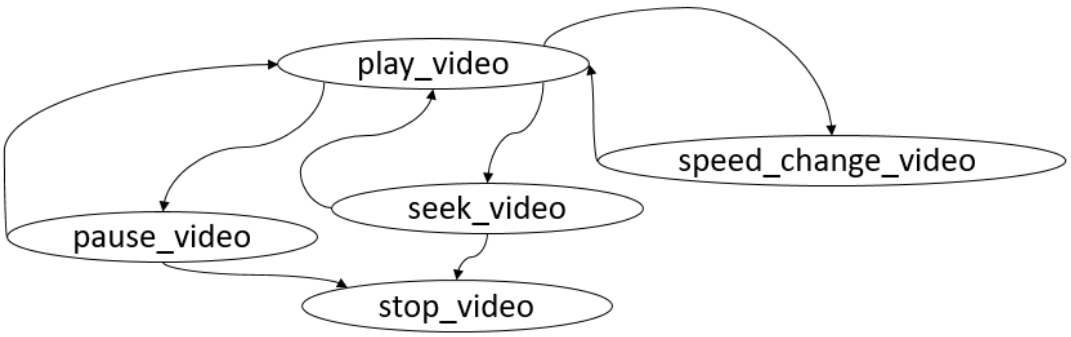

Fig 4.1 The flow of video interaction events.

When users interact with the video, the video system will generate five main events: play_video, pause_video, seek_video, speed_change_video, and stop_video. The play_video event will be generated when users start to play the video. When the video is normally played to the end, the seek_video event will be generated first and then is the stop_video event. When users pause a video, the video system will

generate a pause_video event. When users fast-forward or rewind the video, the video system will produce the speed_change_video event.

Behavioral engagement is referred to learning participation. Thus the stop video event is related to that learners have completed the video watching behavior. Cognitive engagement is referred to learning understanding, thus the pause and seek video events are related to that learners attempt to understand the unclear parts. Emotional engagement is referred to learning affection, thus change video speed event is related to not interest in the content of the video or unconsciously fast-forward/rewind the video.

We used one log event to indicate behavioral engagement. Stop video event: when the video player reaches the end of the video file and play automatically stops.

Cognitive engagement refers to the psychological investment in learning and relates to use self-directed strategies to promote one's understanding [40]. In this study, we measure cognitive engagement by two log events. Pause video event: when a user selects the video player's pause control. Seek video event: when a user selects a user interface control to go to a different point in the video file.

Emotional engagement refers to student attitudes and student interest and values [40]. In this study, we measure emotional engagement by one log event. Speed change video event: when a user selects a different playing speed for the video.

4.2 SPL-Based MOOCs 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.

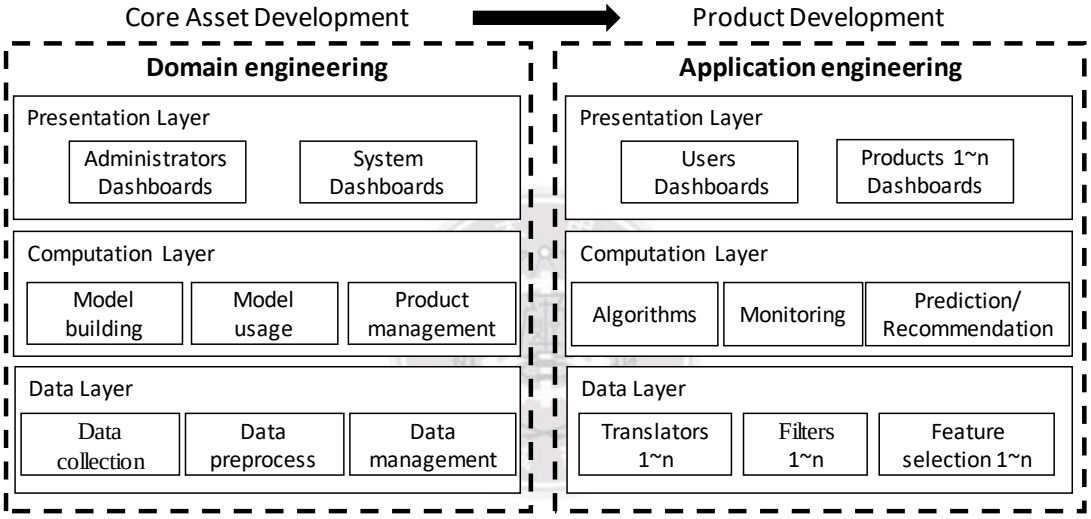

Fig 4.2 SPL-based MOOCs Learning Analytics Framework.

Fig. 4.2 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.

4.2.1 Core Assets Development

4.2.1.1 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.

4.2.1.2 Three Layers of Design of Learning Analytics in Domain Engineering

This divide 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.

4.2.1.3 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 [32].

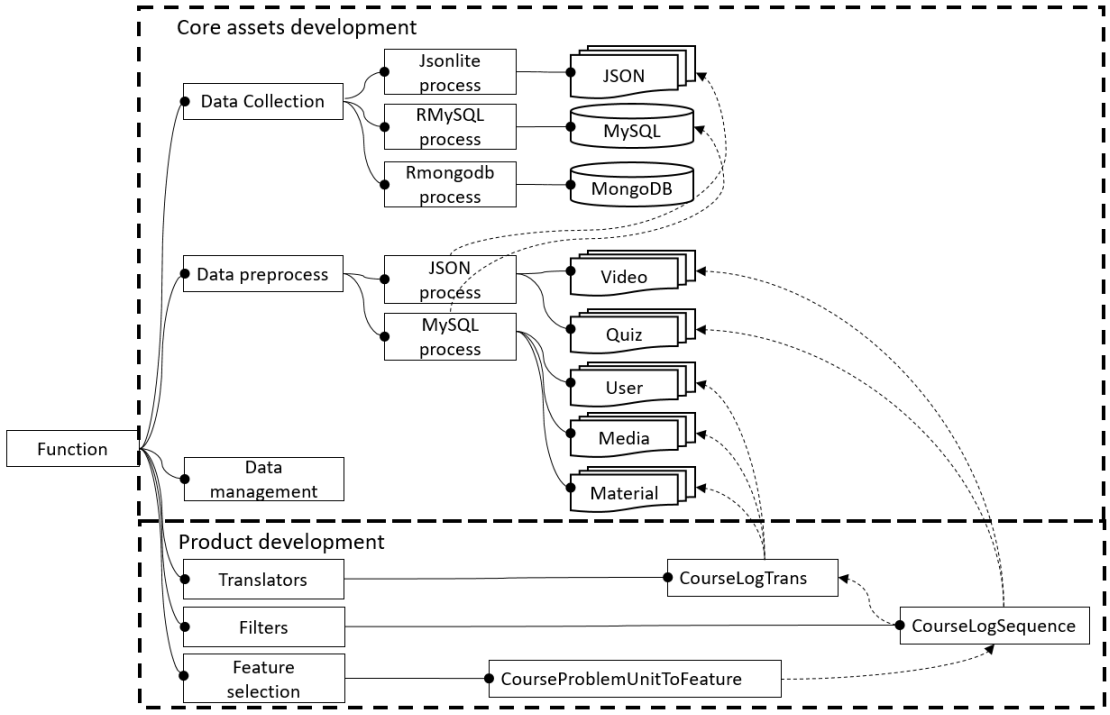

Fig 4.3 Core assets and product development in the Data Layer.

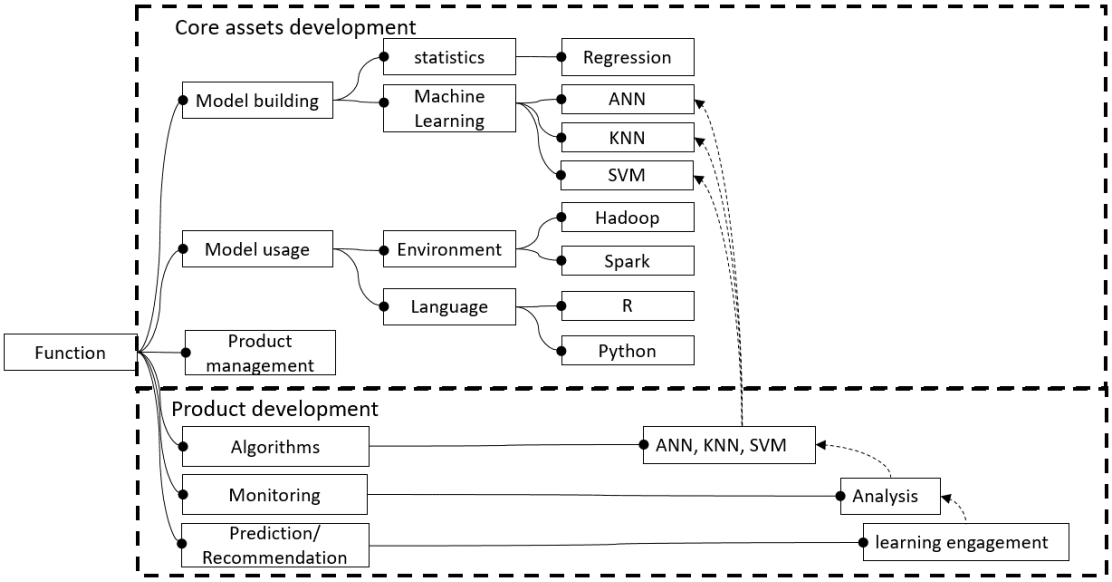

Fig 4.4 Core assets and product development in the Computation Layer.

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. 4.3 and 4.4. 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 [54, 55, 56]. 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. 4.3 and 4.4, 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.

4.2.2 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.

4.2.2.1 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. 4.1. 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.

Since the target MOOC platform in this research is OpenEdu [57], 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 research 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.

4.2.2.2 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.

Table 4.1 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 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 4.1. 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 [58, 59]. After using feature extraction to choose a proper set of features from Table 4.1 for a specified objective, the proposed method performs feature selection to find the best feature sets for prediction model building.

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.

4.2.2.3 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 our 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 includes the learning engagement function based on the Analysis results. This function can produce prediction results and recommend a list of students for further instruction.

4.3 Predicting Learning Outcomes with MOOCs Clickstreams

The video playback events were characterized and divided into eight kinds of feature events according to [39]. The feature event was set as Pl by the start play action of the video (play_video), Pa by the pause action of the video (pause_video), Sf by the forward skipping action of the video (seek_video), and Sb by the backward skipping action of the video (seek_video); the feature event was set as Rf by accelerating the playrate action of the video (speed_change_video) and Rs by decelerating the playrate action of the video (speed_change_video) when the video was played; when the seeking actions of these videos occur within a small time range (<1 second), these seeking events were defined as scroll actions; when the video was played, the feature events were set as Cf and Cb, respectively, by the forward scroll action and the backward scroll action.

The loading action of the video (load_video) sets the feature event as Lo; the ending action of the video (stop_video) sets the feature event as Sp; the subtitle display action of the video (show_transcript) sets the feature event as Sh; and the subtitle hiding of the video (hide_transcript) sets the feature event as Hi.

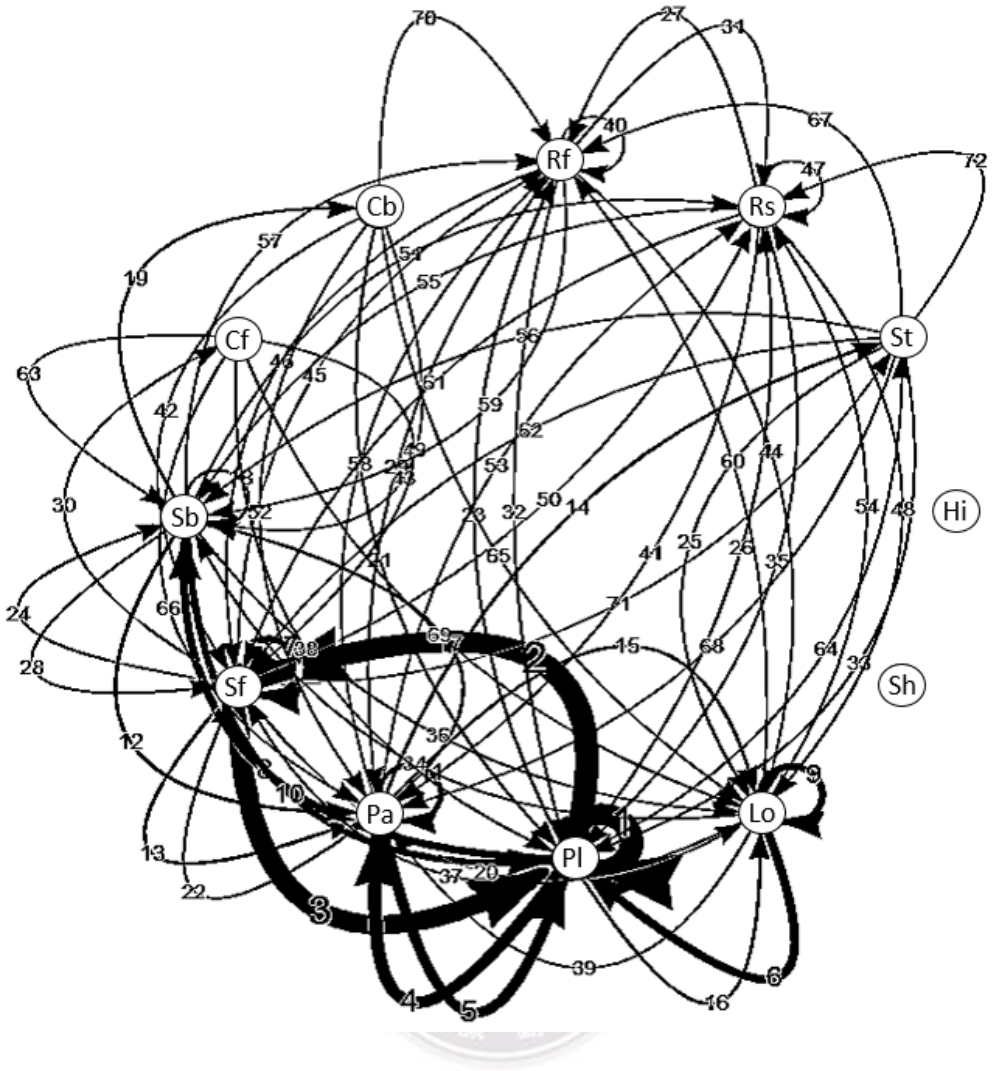

Fig 4.5 OpenEdu video playback event flow.

Table 4.2 The set of feature events derived from OpenEdu video events.

No.	Feature events	OpenEdu video events	Description
1	Hi	hide_transcript	Hide transcript
2	Sh	show_transcript	Show transcript
3	Lo	load_video	Load video
4	Pl	play_video	Play video
5	Pa	pause_video	Pause video
6	Sf	seek_video	Seek Forward
7	Sb	seek_video	Seek Backward
8	Cf	speed_change_video	Scroll Forward (<1 second)
9	Cb	speed_change_video	Scroll Backward (<1 second)
10	Rf	speed_change_video	Ratechange Fast
11	Rs	speed_change_video	Ratechange Slow
12	St	stop_video	Stop Video

Fig. 4.5 observes the occurrence frequency of any two feature events in a course we examined. Since Sh and Hi are less relevant to learning outcomes, they are not included in the observation. For example, in Fig. 4.5, two feature events, Pl and Sf, are generated online. These two feature events have a time-continuous relationship from Pl to Sf under the line number 2. The thicker the line, the greater the number of occurrences. Table 4.4 shows a total of 4,044 times for line number 2. Table 4.2 shows the set of 12 feature events derived from the eight OpenEdu video events.

Table 4.3 Video viewing sequence examples.

OpenEdu video events	Feature event sequence	Number of feature events
load_video, hide_transcript, show_transcript, play_video, speed_change_video, speed_change_video, load_video, hide_transcript, show_transcript, hide_transcript, show_transcript, load_video, play_video, speed_change_video, speed_change_video, speed_change_video	Lo Pl Rs Rs Lo Lo Pl Rf Rf Rs	10
load_video, hide_transcript, show_transcript, play_video, show_transcript, load_video, hide_transcript, show_transcript, load_video, hide_transcript, play_video, pause_video, stop_video, play_video, seek_video, play_video, seek_video, play_video, seek_video, play_video, pause_video	Lo Pl Lo Lo Pl Pa St Pl Sb Pl Sb Pl Sb Pl Pa	15
hide_transcript, show_transcript, load_video	Lo	1
load_video, hide_transcript, show_transcript, play_video, pause_video, stop_video, load_video, hide_transcript, show_transcript, hide_transcript, show_transcript, load_video, play_video, seek_video, play_video, seek_video, play_video, play_video, seek_video, seek_video, seek_video, play_video, seek_video, play_video, play_video, seek_video, seek_video, play_video, play_video, seek_video	Lo Pl Pa St Lo Lo Pl Sf Pl Sf Pl Pl Sf Sf Pl Sf Pl Pl Sf Sf Pl Pl Sf	23
show_transcript, load_video, hide_transcript, play_video, pause_video, play_video, pause_video, play_video, pause_video, play_video, pause_video, stop_video	Lo Pl Pa Pl Pa Pl Pa Pl Pa St	10

In order to analyze the sequence of video viewing behaviors, we recorded the same session number upon learners opening a browser for networking when they

logged in to view course videos and generate learning events. The process of events is recorded chronologically. Such a record was the behavior sequence of a learner in viewing a certain video. We studied event items in the behavior sequence to remove event items that are not related to learning, such as show_transcript and hide_transcript, and convert them into feature sequences. The feature sequences form a specific combination of features for the learner to view the continuity of the course unit video. The features combination is a continuous sentence composed of two specific characters, and then the word exploration is used to score the sentence similarity. Therefore, the content of each feature sequence is N-gram processed to observe the frequency and percentage of similarity occurrence. Table 4.3 sets out the case studies on five example records, and each of them is a video-clicking operating record of each learner under the same session networking condition in viewing a certain course video. The No. 1 record is the learner's viewing behavior sequence and ten feature events in total have been obtained by chronologically processing the 16 video events clicked by the learner.

Table 4.4 Top 10 frequencies of feature event sequences for 2-grams.

No.	2-grams	frequency	proportion
1	Pl Pl	5366	0.2012
2	Pl Sf	4044	0.151631
3	Sf Pl	3644	0.136633
4	Pl Pa	3133	0.117473
5	Pa Pl	2978	0.111661
6	Sf Sf	1288	0.048294
7	Lo Pl	1129	0.042332
8	Pl Sb	955	0.035808
9	Sb Pl	694	0.026022
10	Lo Lo	596	0.022347

After applying the N-gram package of R, this study observed that two feature events (2-grams) occurred in a total of 72 combinations, of which the combinations with the top 10 highest frequencies were PlPl, PlSf, SfPl, PlPa, PaPl, SfSf, LoPl, PlSb, SbPl, and LoLo, in sequence, as shown in Table 4.4. The three feature events (3-grams) occurred in a total of 407 combinations, of which the combinations with the top 10 highest frequencies were Pl Pl Pl, Pl Sf Pl, Pl Pa Pl, Pa Pl Pa, Sf Pl Sf, Pl Pl Sf, Sf Pl Pl, Pl Sf Sf, Sf Sf Pl, and Pa Pa Pa, in sequence (Table 4.5). In addition, four feature events (4-grams) occurred in a total of 1,508 combinations, of which the combinations of the top 10 highest frequencies were Pl Pl Pl Pl, Pa Pl Pa Pl, Pl Pa Pl Pa, Pl Sf Pl Sf, Sf Pl Sf Pl, Sf Pl Pl Sf, Pl Sf Sf Pl, Pl Pl Sf Pl, Pa Pa Pa Pa, and Sf Sf Pl Pl, in sequence (Table 4.6). Based on [16], we found that the length of 2-grams and 3-grams is too short to manually identify learning behaviors for video clickstreams. Therefore, 4-grams is used in our analysis, as determined empirically.

Table 4.5 Top 10 frequencies of feature event sequences for 3-grams.

No.	3-grams	frequency	proportion
1	Pl Pl Pl	3664	0.142396
2	Pl Sf Pl	2801	0.108857
3	Pl Pa Pl	2580	0.100268
4	Pa Pl Pa	2469	0.095954
5	Sf Pl Sf	2432	0.094516
6	Pl Pl Sf	1184	0.046015
7	Sf Pl Pl	839	0.032607
8	Pl Sf Sf	790	0.030702
9	Sf Sf Pl	729	0.028332
10	Pa Pa Pa	483	0.018771

Table 4.6 Top 10 frequencies of feature event sequences for 4-grams.

No.	4-grams	frequency	proportion
1	Pl Pl Pl Pl	3484	0.139678
2	Pa Pl Pa Pl	2313	0.092731
3	Pl Pa Pl Pa	2230	0.089404
4	Pl Sf Pl Sf	2156	0.086437
5	Sf Pl Sf Pl	1829	0.073327
6	Sf Pl Pl Sf	728	0.029187
7	Pl Sf Sf Pl	677	0.027142
8	Pl Pl Sf Pl	644	0.025819
9	Pa Pa Pa Pa	464	0.018602
10	Sf Sf Pl Pl	414	0.016598

Table 4.7 Grouping clickstream feature sequences to form behavioral actions.

No.	Behavioral actions	Clickstream feature sequences
1	Rewatch	SbPl**, *SbPl*, **SbPl, PlSb**, *PlSb*, **PlSb, Sb*Pl*, *Sb*Pl Pl*Sb*, *Pl*Sb
2	Skipping	SfSf**, *SfSf*, **SfSf, Sf*Sf*, *Sf*Sf
3	Fast Watching	PlRf**, *PlRf*, **PlRf, RfRl**, *RfRl*, **RfRl, Pl*Rf*, *Pl*Rf, Rf*Pl*, *Rf*Pl
4	Slow Watching	Pl*Rs*, *Pl*Rs, Rs*Pl*, *Rs*Pl
5	Clear Concept	SbCb**, *SbCb*, **SbCb, Sb*Cb*, *Sb*Cb
6	Checkback Reference	SbSb**, *SbSb*, **SbSb, Sb*Sb*, *Sb*Sb
7	Playrate Transition	RfRf**, *RfRf*, **RfRf, Rf*Rf*, *Rf*Rf, RfRs**, *RfRs*, **RfRs, Rf*Rs*, *Rf*Rs, RsRs**, *RsRs*, **RsRs, Rs*Rs*, *Rs*Rs, RsRf**, *RsRf*, **RsRf, Rs*Rf*, *Rs*Rf

*: don't care eigenvalue mode; **: two consecutive don't care eigenvalue mode.

According to [60], the behavioral actions of the video viewing sequence can be divided into seven types: Rewatch, Skipping, Fast Watching, Slow Watching, Clear Concept, Checkback Reference, and Playrate Transition. Next, the above twelve feature events of Table 4.2 are used to define each type of behavioral actions, provided the said behavior conforms to one of the video playback feature sequences. Therefore, grouping clickstream sequences to form higher-level categories, instead of raw clicks, better exposes the browsing pattern of leaners. Due to the use of the fixed sequence mode, such as the top 'k' most frequent 4-grams, the frequency in full

coincidence with the feature sequences is very low. Therefore, we use the *: don't care eigenvalue mode to group clickstream feature sequences to form behavioral actions, as shown in Table 4.7. For example, Rewatch is formed with the combinations of Seek Backward and Play, as well as two occurrences of don't care [26, 44].

Table 4.8 Feature table of course unit activities.

No.	Feature	Descriptions
1	unit_num	Total number of Login course units
2	video_num	Total number of viewing unit videos
3	sess_num	Total number of online videos viewing sessions
4	Rewatch	Total number of clickstream feature sequence occurrences of Rewatch
5	Skipping	Total number of clickstream feature sequence occurrences of Skipping
6	Fast Watching	Total number of clickstream feature sequence occurrences of Fast Watching
7	Slow Watching	Total number of clickstream feature sequence occurrences of Slow Watching
8	Clear Concept	Total number of clickstream feature sequence occurrences of Clear Concept
9	Checkback Reference	Total number of clickstream feature sequence occurrences of Checkback Reference
10	Playrate Transition	Total number of feature sequence occurrences of playback speed change behavior
11	exam_num	Total number of tests available
12	prom_num	Total number of answers to a test
13	all_attempts	Total number of attempts to respond to a test
14	unit_score	Total score of a test in a course unit

The resulting feature records of video watching statistics and test results are merged based on the test unit of the course to record their answers and scores. If a video is not followed by a test in the current learning unit, its viewing statistics will be recorded in the next test unit, which can be used as a predictive feature of learning engagement. The feature items include the number of entries to the course unit, the number of online videos played, the number of playbacks, load times, play times,

pause times, stop times, seek times, speed_change times, Rewatch, Skipping, Fast Watching, Slow Watching, Clear Concept, Checkback Reference, Playrate Transition, the number of tests used, the number of tests answered, the number of tests tried, unit test scores, final test scores, course scores, and course assignment scores. Therefore, the generated features of course unit activities have a total of 85 feature items. Through the feature selection function, we selected 14 feature values for machine learning model building and prediction, as shown in Table 4.8. Note that video viewing and unit test activities are included in the feature set. This is because we found that some learners did not take the test after viewing the videos in a course unit. On the other hand, some learners took the test without viewing videos.

Chapter 5 Experiments

In this chapter, we present several experiments to demonstrate that our implementations in Chapter 4 are feasible and satisfactory in meeting our research objectives. First, the research environment is explained in Section 5.1. Then, the experiments for applying learning analytics to deconstruct user engagement in Phase 1 is described in Section 5.2. The experiment for Phase 2 is reported in Section 5.3 to show the use of the SPL-Based MOOCs Learning Analytics Framework. Finally, the experiment for predicting learning outcomes with MOOCs clickstreams in Phase 3 is described in Section 5.4.

5.1 Environment

Table 5.1 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

To verify that the proposed SPL-based Analytics framework is feasible, we 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 5.1, open source tools were used for development, and the function set used is shown in Table 5.2.

Table 5.2 Function set list.

Name	Command	Description
nnet	ann	Feed-Forward Neural Networks and Multinomial Log-Linear Models
ISLR	knn	k-Nearest Neighbour 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 dbDisconnect	Database Interface and 'MySQL' Driver for R

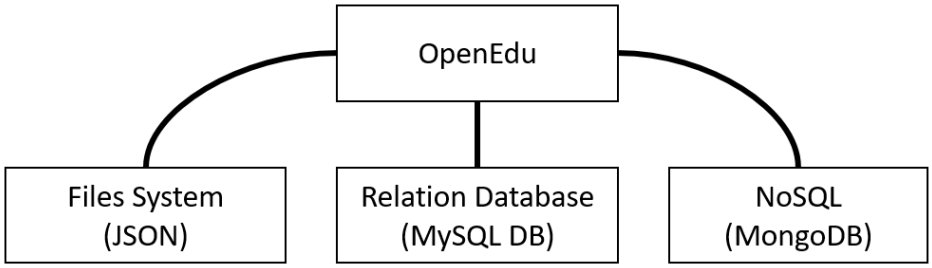

Fig 5.1 OpenEdu data architecture.


```
{
  "event_source": "browser",
  "event": "{\"id\":\"i4x-BerkeleyX-Stat_2_1x-video-
58424ad2f75048798b4480aa699cc215\",\"currentTime\":243,\"code\":\"iOOYGgLADj8\"}",
  "time": "2014-12-23T14:26:53.723188+00:00",
  "referer":
"http:\/\/localhost:8001\/container\/i4x:\/\/edX\/DemoX\/vertical\/69dedd38233a46fc89e4d7b5e8da1bf4?action=new",
  "accept_language": "en-US,en;q=0.8",
  "event_type": "play_video",
  "session": "11a1111111a1a1a1aa1a11a1a1111111",
  "agent": "Mozilla\/5.0 (Windows NT 6.1; WOW64) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/39.0.2171.95
Safari\/537.36",
  "page":
"https:\/\/courses.edx.org\/courses\/BerkeleyX\/Stat_2.1x\/1T2014\/courseware\/d4ff35dabfe64ed5b1f1807eb0292c73\/bd34
3b7dcb2c4817bd1992b0cef66ff4\/",
  "username": "AAAAAAAAAA",
  "ip": "123.123.123.123",
  "context": {
    "org_id": "BerkeleyX",
    "path": "\/event",
    "course_id": "BerkeleyX\/Stat_2.1x\/1T2014",
    "user_id": 99999999
  },
  "host": "courses.edx.org"
}
```


Fig 5.2 OpenEdu JSON of tracking log.

MOOCs platform learning records are OpenEdu MOOCs platform data stored in MySQL and MongoDB (Fig. 5.1), while the Tracking Log is stored on the server end in the JSON files format. The contents of the MySQL relational database include user profile, course records, course basic data, etc. The contents stored in the MongoDB NoSQL database include course discussion area content, course videos, course exercises, etc. The Tracking Log records the user's behavior on the website, where the records are distinguished by events and have a time stamp. Fig. 5.2 shows the record contents in JSON format when videos are played for students. All users' operating activities on a website were recorded. Such records were classified by events and attached with a timestamp. The events included video playback events, discussion forum events, answering events, and browsing website events. Table 5.3 sets out the description of each field of JSON contents, including username, session, ip, event source, event type, event, agent, page, time, and context. This study conducts follow-up studies with the data taken from viewing videos. The play action includes six events: load_video, play_video, pause_video, seek_video, speed_change_video, and stop_video [22].

Table 5.3 Field description of student learning behavior in the Tracking Log.

No	Field	Type	Description
1	username	String	User id
2	session	String	Network session id
3	ip	String	IP information of the user's network
4	event source	String	Category of event source
5	event type	String	Event type
6	event	JSON	Detail field of the event
7	agent	String	Terminal information of the user
8	page	String	Web page of the event
9	time	Timestamp	Time of the event
10	context	JSON	Context of the event

5.2 Applying learning analytics to deconstruct user engagement

In this study we use log-data from 155 MOOCs in OpenEdu platform. The courses ranged from 2014 to 2016. Eighty-four courses which provided quizzes and contained five video events (play video, pause video, stop video, seek video, speed change video) were considered in this study. 6433 students registered these 84 courses and log-data on 2697 students who attempted at least one quiz, and contained five video events after registration was used in this study.

In this research, the behavioral engagement was measured by the times of videos students stopped each course. The cognitive engagement was measured by the number of pauses, and seeking in videos watched in each course. The emotional engagement was measured by the number of speed changes in videos watched in each course. In addition, we measured learning outcomes by students' total quiz scores (the sum of scores a student got on each quiz he/she attempted each course). Also, we generated an "is_passed" binary variable from calculating total quiz scores (if a student's total quiz score was greater than 60, is_passed was calculated as passed; otherwise, it was failed). 1125 students were labeled as passed, and 1572 students were labeled as not passed.

We applied correlation to investigate whether there are relationships between three engagement components and total quiz scores. A K-means technique was employed to partition the students according to three engagement components. We also employed multiple linear regression techniques to predict student scores by video event variables. Furthermore, three classification methods (Support vector machine, Random forest, Artificial neural network) were performed and compared to predict "is_passed" variable. The 10-fold cross-validation is used to assess the accuracy and validity of classification models.

To further understand user engagement in MOOCs, this study employed Lag

Sequential Analysis (LSA) to discover the difference of behavioral patterns of passed and failed students in MOOCs. LSA is used to test the statistical significance level of sequential correlation among video event variables [54]. The statistics of LSA involves a series of steps. The first step is to arrange the video watching events in chronological order. The second step is to conduct the following matrix calculations: (1).Sequential frequency transfer matrix; (2).Condition probability matrix; (3).Expected-value matrix. The third step is to calculate Z-scores using the calculated matrices. The sequential behavioral patterns with a Z-score higher than 1.96 ($p < 0.05$) were considered as significant. More importantly, this study only chooses the most popular video in each course to conduct LSA.

5.2.1 Correlation between video events and quiz scores.

As shown in Table 5.4, Spearman's rank correlations were calculated.

Table 5.4 Correlation between video events and quiz scores.

	Components	**Events**	***R***	***P***
Scores	Behavioral	Stop video	0.36	.00
	Cognitive	Pause video	0.42	.00
		Seek video	0.37	.00
	Emotional	Speed change video	0.17	.00

From Table 5.4, we confirmed significant and high overall positive relation between the behavioral engagement and quiz scores. Similarly, there was a significant and very high positive correlation between the cognitive engagement and quiz scores. Interestingly, we found that there was a significant and moderate positive correlation between the emotional engagement and quiz scores.

5.2.2 K-means clustering analysis

To better understand the relationship between three engagement components, we used K-Means to partition the students into three clusters according to their engagement variables. According to the results of practical experiment tests, the three groups will be more obvious to find the learner's behavior participation model. The cluster model revealed that Cluster 1 to 3 had 1257, 640, 800, and respondents, respectively. To draw the result of clustering, a principal component analysis was performed to reduce the dimensionality of features. As shown in Fig. 5.3, the result of K-Means clustering exists three clusters.

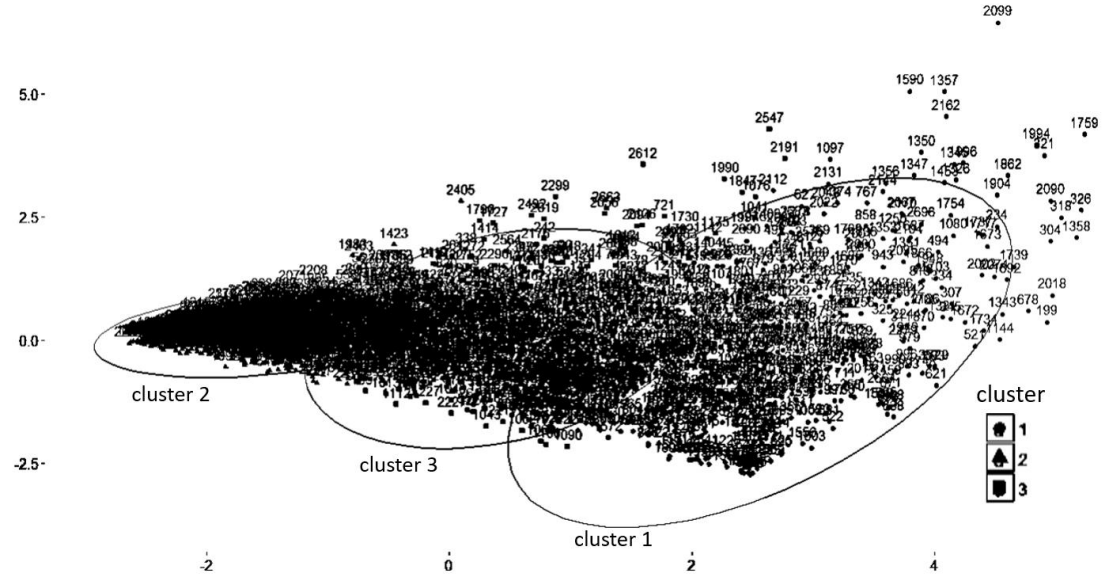

Fig 5.3 The result of K-Means clustering.

Table 5.5 indicates that Cluster 3 got the highest scores, the following was Cluster 2, and Cluster 1 got the worst scores. The frequency of video events, interestingly shows a similar pattern. Cluster 3 got the highest frequency in all video events. On the contrary, Cluster 1 got the lowest frequency in all video events. The

findings may imply that high scoring students have a preference to watch videos at their own pace. They are frequently to stop, pause and seek the video play. However, the low score students are not.

Table 5.5 The means of video events and quiz scores in three clusters.

Cluster	Stop	Pause	Seek	Speed change	Score
1	0.1	0.14	0.11	0.06	0.33
2	0.34	0.39	0.27	0.12	0.54
3	0.65	0.76	0.44	0.17	0.71

5.2.3 Multiple linear regression analysis

A multiple linear regression was undertaken to examine the variance in students' total quiz scores. Four predictors were loaded into the model using the Enter method. Table 5.6 shows that the model was able to explain 22.7% of the sample outcome variance (Adj. R2 = .226), which was found to significantly predict the outcome, $F(4, 2692) = 197.767$, $p < .001$. Three of the predictor variables significantly contributed to the model. High frequencies of pausing video, seeking, and stopping video were related to higher quiz scores. The frequency of changing video speed did not contribute to variance. There was a medium effect size ($d = 0.29$).

Table 5.6 Multiple linear regression analysis of quiz scores.

	R^2	**Adj.R^2**	**F**	**P**	**Constant**
Model	.227	.226	197.767	<.001	.276
Predictor variable			**Gradient**	**t**	**p**
Stop video			.264	8.58	.000
Pause video			.143	4.08	.000
Seek video			.316	10.23	.000
Speed change video			.006	0.11	.912

5.2.4 Classification analysis

Three classification methods (Support vector machine, Random forest, Artificial neural network) were performed and compared to predict is_passed variable. The Confusion matrices of three classification methods are shown in Table 5.7, 5.8, and 5.9. The elements in the confusion matrix indicate the correctly and incorrectly classified data for the passed and failed classes. These matrices are used to evaluate the performances of three classification methods. The overall classification precision is high (>70%). The overall recall rate of the failed class is high (>79%). However, the recall rates of the passed class for three classification method varied. The Random Forest method gets a lower recall rate of the passed class (56.44%) and the ANN method achieves highest one (66.93%). Moreover, the accuracy of SVM, Random Forest, and ANN are 73.79%, 73.82%, and 74.34%, respectively. The ANN method gets the highest accuracy. In summary, the ANN method outperforms the SVM and Random Forest methods.

Table 5.7 Confusion matrix of SVM results.

SVM	true failed	true passed	class precision
pred. failed	**1281**	416	75.49%
pred. passed	291	**709**	70.90%
class recall	81.49%	63.02%	

Table 5.8 Confusion matrix of Random Forest results.

Random Forest	true failed	true passed	class precision
pred. failed	**1356**	490	73.46%
pred. passed	216	**635**	74.62%
class recall	86.26%	56.44%	

Table 5.9 Confusion matrix of ANN results.

ANN	true failed	true passed	class precision
pred. failed	**1252**	372	77.09%
pred. passed	320	**753**	70.18%
class recall	79.64%	66.93%	

5.2.5 Sequential patterns of video watching behavior for the passed and failed users

Table 5.10 presented the Z-scores of all users. Table 5.11 and 5.12 only presented the Z-scores of the passed users and the failed users. If the Z-score is more than 1.96, it illustrates p is less than 0.05. In other words, the video watching behavior from the row to the column is significant in sequence. Take Table 5.11, for example, the Z-score of 'play' row and 'seek' column is more than 1.96, and it indicates that the behavioral sequence from 'play' event to 'seek' event ('play' -> 'seek') reaches continuity significantly. Based on the calculated Z-scores, the sequential patterns of the all users (Fig. 5.4), the passed users (Fig. 5.5) and the failed users (Fig. 5.6) are displayed. In these figures, each node represented a video event, and the nodes connected with solid lines and arrow suggested the sequential relationships between video events reached statistical significance.

Table 5.10 Adjusted residuals table (Z-score) of video watching behavior for all users.

	play	seek	pause	Stop	speed
play	-110.85	24.38*	95.82*	-0.34	11.42*
seek	44.89*	41.65*	-60.44	-44.52	-30.08
pause	89.49*	-60.71	-60.78	59.74*	-19.20
stop	17.41*	-25.21	19.92*	-11.93	-7.39
speed	-24.94	-20.96	9.50*	1.37	94.48*

* $p<0.05$

Table 5.11 Adjusted residuals table (Z-score) of video watching behavior for passed users.

	play	seek	pause	stop	speed
play	-77.91	25.89*	57.80*	2.02*	7.99*
seek	44.6*	13.62*	-39.72	-30.59	-19.47
pause	47.39*	-33.55	-38.11	40.41*	-13.04
stop	13.99*	-19.46	15.36*	-9.86	-5.96
speed	-17.24	-13.03	8.01*	-1.03	60.16*

* $p<0.05$

Table 5.12 Adjusted residuals table (Z-score) of video watching behavior for failed users.

	play	seek	pause	stop	speed
play	-79.51	10.46*	76.91*	-2.95	8.1*
seek	21.38*	42.17*	-45.82	-31.99	-22.89
pause	77.59*	-51.27	-47.41	44.44*	-14.08
stop	9.78*	-15.45	13.09*	-7.24	-4.49
speed	-18.12	-16.38	5.64*	2.93*	72.99*

* $p<0.05$

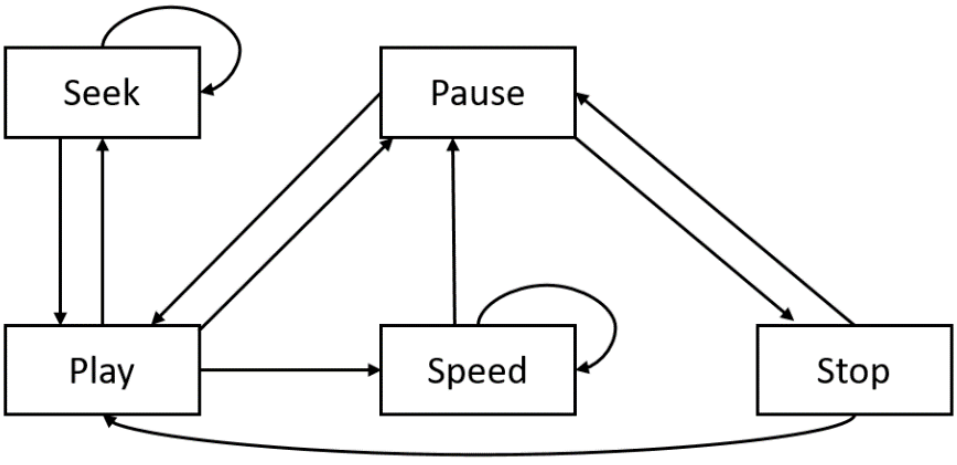

Fig 5.4 The video watching behavior of all users.

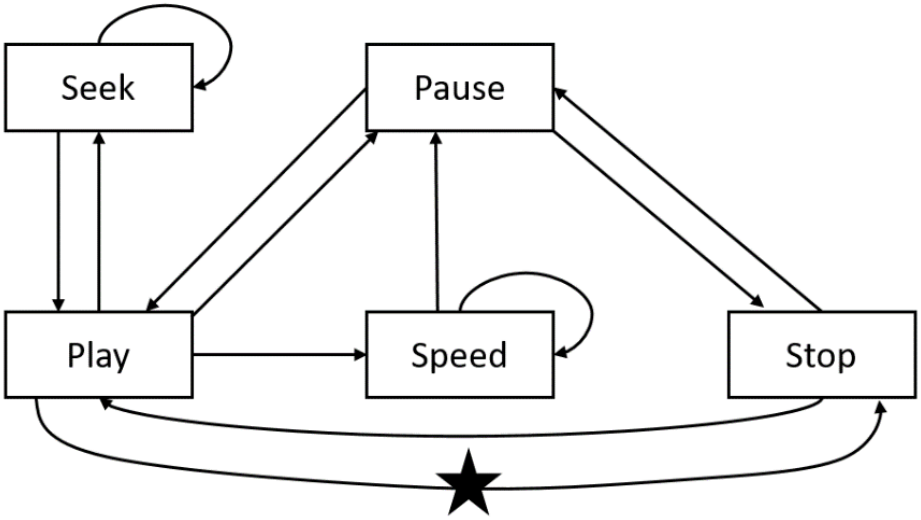

Fig 5.5 The video watching behavior of the passed users.

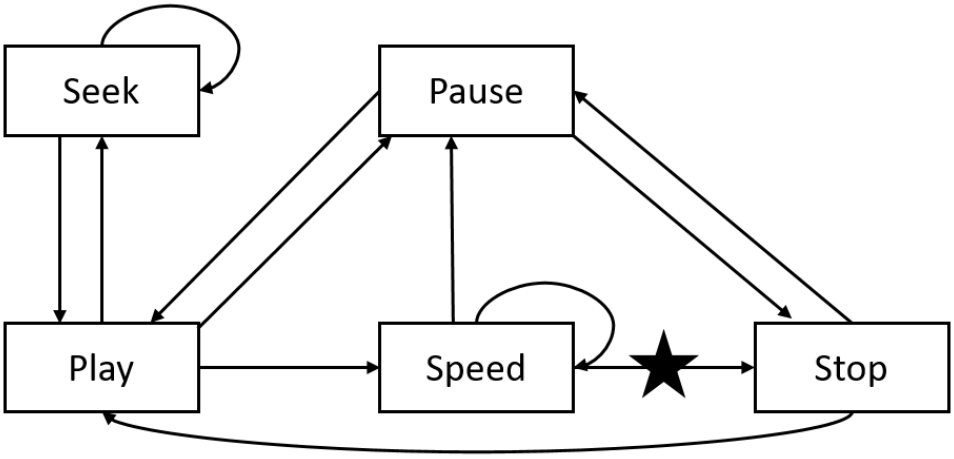

Fig 5.6 The video watching behavior of the failed users.

In Fig. 5.4, the video watching behavior of all users reveals that these learners tended to play a video and seek the video (play → seek; seek → play), or pause the video (play → pause; pause → play). They may also play a video and change the video speed (play → speed), or pause the video after changing the video speed (speed → pause). Moreover, they may also stop a video, and pause the video (stop → pause; pause → stop), or play the video (stop → play). The overall pattern may

exhibit that after playing the video, users tend to either (1) continually seek the video to go to a different point in the video file, or (2) pause the video or change the video speed, and then finally stop the video.

The Fig. 5.5 and 5.6 of the sequential patterns can be employed to further compare the differences between the passed- or the failed- users. The asterisks in Fig. 5.5 note the specific significant relationships observed for the passed users, which are not found in the failed users. Likewise, the asterisks in Fig. 5.6 are the significant relationships found in the failed users that are not significantly observed for the passed users.

As for the passed users (see Fig. 5.5), they tended to play a video through to the end, and the player stopped the video automatically (play → stop). Contrary to the passed users, the failed users tended to slow down or speed up a video through to the end, and then the player stopped the video automatically (speed → stop) (Fig. 5.6).

5.3 SPL-Based MOOCs Learning Analytics Framework

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.

5.3.1 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 4.1 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 5.13.

Table 5.13 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.

5.3.2 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.

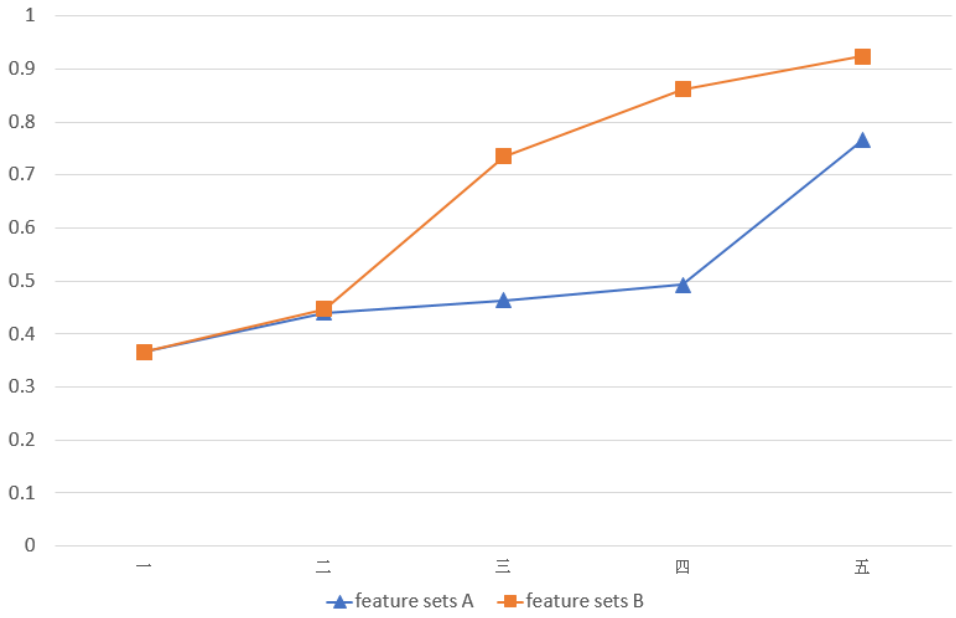

Fig 5.7 Weekly prediction accuracy

Fig. 5.7 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.

5.4 Predicting Learning Outcomes with MOOCs Clickstreams

As the course contents of the MOOCs platform are very diversified, a foundation course, which had been offered at least three times, was selected to conduct analysis. The first class lasts for 6 weeks, contains 55 videos, and has a total of 532,579 learning process records. A total of 590 students took part in the course, of whom 264 obtained the certificate, while 327 failed to obtain the certificate. As mentioned at the end of the previous section, we found that there were 3,706 records of learning activities in terms of course unit. (a) Among the 3,706 records, there were 511 of them without taking unit tests. (b) For those taking the unit tests, there were 538 of them without viewing any video in a course unit. Therefore, it is necessary to include both the clickstream of the video viewing and unit test score in the feature set. The second class has a total of 256,000 learning process records; a total of 346 students participated in the course, of whom 137 obtained the certificate, while 209 failed to obtain the certificate. The third class has a total of 137,348 learning process records; a total of 427 students participated in the course, of whom 57 obtained the certificate,

while 370 failed to obtain the certificate. We will combine the learning process records of the first two semesters to build prediction models and select the best model to verify the prediction accuracy with the data from the third semester.

As mentioned before, the N-gram method of R language is used with n = 4 and the sequence analysis results show that a total of 1,508 combinations occurred in the four feature events (4-grams) from the data of the second class, of which the combinations of the top 10 highest frequencies are presented in Table 4.6. Then, the don't care eigenvalue mode is used to form the seven behavioral actions of learners' learning engagement, as shown in Table 4.7. Table 4.8 presents the feature set of course unit activities for building prediction models.

First, when the KNN method of R language is used, the library(ISLR) suite should be loaded beforehand using the knn() function, including 70% of the feature data set for training, 30% of the feature data set for testing, and the real classification factors are pass (1) and fail (0) of the course, where the K value (# of neighbors) is calculated as the square root of the number of click counts, and the accuracy is 0.8624535.

As the accuracy of the KNN model is poor, we use the SVM method next. The library(e1071) suite should be loaded beforehand in R language, and svm() is used to train the classification model of SVM, including 70% data for training, 30% data for testing, and the target values are pass (1) and fail (0) as well. The accuracy of the built model is 0.9442379.

For further improvement, we use the ANN method of R language and the library(nnet) suites should be loaded beforehand using the ann() function, including 70% data for training, 30% data for testing, and the target values are pass (1) and fail (0). The number of units in the hidden layer was set from 1 to 10, the parameter of the specific gravity attenuation was 0.001, and the maximum number of repetitions was

1000. When the numbers of units in the hidden layer were 1 to 5, the highest accuracy of the model was 0.9516729. The ANN model has the highest average accuracy as well.

We also performed the experiments using 80% of the feature data set for training and 20% of the feature data set for testing. It is found that the accuracy of the 80–20% partition is better than that of the 70–30% partition. For example, the third column of Table 5.14 shows that the highest accuracy of the ANN model for the first two classes offered is 0.955307263 when the number of hidden layers is 2 or 3. Then, based on the best ANN model built by using the first two classes data set, we use the third class data set as testing data to find the prediction accuracy, as shown in the fourth column of Table 5.14. We can see that the average prediction accuracy of the third class data (0.977578475) is slightly better than the average model accuracy of the first two classes of data (0.950837989). The result is better than we expected, since the two classes were offered in different semesters.

Table 5.14 Predictive accuracy of ANN models.

Model	Size	Model accuracy of the first two classes offered	Prediction accuracy of the third class offered
ANN	1	0.94972067	0.97309417
	2	0.955307263	0.986547085
	3	0.955307263	0.977578475
	4	0.944134078	0.977578475
	5	0.94972067	0.97309417
Average		0.950837989	0.977578475

Based on the built models, we can also make weekly predictions. Thus, the weekly tutoring list of students was provided for teachers to supervise students' learning progress. There were 313 students who needed tutoring in the first week, 313 in the second week, 311 in the third week, 305 in the fourth week, and 297 in the fifth week.

In addition, we would like to generalize the research result and show that our approach can be applied to any new courses. We selected another MOOC course that lasts for 9 weeks, contains 95 videos, and has a total of 447,133 learning process records. A total of 977 students took part in the course, of whom 86 obtained the certificate, while 891 failed. First, 70%/30% partitions of course data into training/verification data were performed. The KNN, SVM, and ANN methods were used to generate prediction models and their model accuracies are 0.8787879, 0.9218182, and 0.9318182, respectively. ANN is still the best. Using 80%/20% partitions, the model accuracy of ANN increases to 0.9431818. This is consistent with the previous result of our approach.

Table 5.15 Three video classes and their combinations.

No.	Combinations of video classes	Number of videos	Total number of video clicking records
1	All of the three classes	55	13,124
2	Theoretical	16	5641
3	Experimental	25	5713
4	Analytic	14	1615
5	Theoretical and experimental	41	11,352
6	Theoretical and analytic	30	7176
7	Experimental and analytic	39	7248

To make further improvements and provide more information for teachers, we classify course videos based on their contents and perform additional analysis. Course videos were classified into three classes: theoretical videos, experimental videos, and analytic videos. Types of video viewing behaviors were compared through cross-validation and analysis of the frequent values distribution diagram. There were indeed different significant features with respect to learners passing and not passing the course. It was found that students did not view and click all the videos in their

video viewing model according to analysis of course learning records and course unit videos clicked by students. As such, we examined more details on these three types of video content, and seven types of models were formed through combinations, as shown in Table 5.15. For example, there were a total of 55 videos with respect to the whole course in the overall class of all three of them, with a total of 13,124 video clicking records.

Table 5.16 Accuracy of the Top 13 ANN models.

No.	Combinations of video classes	Size of hidden layer	Accuracy
1	Overall or All of the three classes	1	0.960452
2	Theoretical and experimental	1	0.954802
3	Theoretical and experimental	2	0.949153
4	Theoretical and analytic	2	0.931429
5	Overall or All of the three classes	2	0.926554
6	Overall or All of the three classes	3	0.926554
7	Theoretical	2	0.925714
8	Theoretical and analytic	1	0.925714
9	Experimental	2	0.921429
10	Experimental and analytic	3	0.921429
11	Experimental and analytic	1	0.914286
12	Experimental	1	0.9
13	Experimental and analytic	2	0.9

There were 21 prediction accuracy items of Size 1~3 of ANN implemented based on the seven types of models. In total, 13 items falling within the scope of ANN Size < 4 and Accuracy > 0.9 have been extracted (Table 5.15). Library (stats) suite must be loaded first in R language. The contingency table analysis method was implemented by using table(). We use Overall or All of the three classes as the First category, and take the most accurate Size of the hidden layer as 1 (Table 5.16). Using the First category of this item and taking the Size of the hidden layer as 1, the tandem analysis is performed on different Second categories, and the prediction accuracy is determined by using the contingency table, resulting in the top 13 items of Table 5.17.

Therefore, among those generated with the best accuracy in Table 5.17, we find that there are 12 items above the minimum standard (>=75%), and the ratio accounts for 92%. There are 10 items above the winning bid of median standard (>=80%), with a ratio of 46%. There are four items above the best standard (>=90%), and the ratio accounts for 31%, as shown in Table 5.18. The results are in line with our purpose.

Table 5.17 Result of the Top 13 models in terms of accuracy.

First category	Size of hidden layer of the first category	Second category	Size of hidden layer of the second category	accuracy
Overall or All of the three classes	1	Theoretical and experimental	3	0.988764045
		Theoretical and analytic	3	0.91011236
		Theoretical	2	0.904494382
		Theoretical	1	0.904494382
		Theoretical and analytic	1	0.887640449
		Theoretical and analytic	2	0.887640449
		Experimental and analytic	2	0.882022472
		Experimental	2	0.876404494
		Experimental and analytic	3	0.859550562
		Theoretical	3	0.853932584
		Analytic	2	0.752808989
		Analytic	3	0.752808989
		Analytic	1	0.730337079

Table 5.18 Summary of the three-level Base Evaluation from Table 5.17.

Levels	Range	#s within the range	Ration within the range
Best	Accuracy >= 90%	4	31%
Median	Accuracy >= 80%	10	46%
Minimum	Accuracy >= 75%	12	92%

It was found through the matching test that ANN size = 1 in the overall category was matched with ANN size = 1 in the theoretical and experimental category (see the first entry of Table 5.17), and the prediction accuracy values were close (see the first two entries of Table 5.16).

Chapter 6 Conclusions and Future Works

6.1 Summary

In the Phase 1, this study applied learning analytics to deconstruct user engagement by using log data of MOOCs. In other words, the engagement was first deconstructed into three components (behavioral engagement, cognitive engagement, and emotional engagement). And then, the importance of each component was evaluated by examining their helpfulness in predicting grades. The results of both correlation and clustering analysis of the three components and quiz scores shows that high scoring students had a preference to stop, pause, seek, and speed up/down the video play at their own pace. More specifically, they were more frequent in all of the three components than the low score students. For instructors, the findings may imply that they can use any of the three components to initially estimate student engagement. For system developers, the results indicate that it is important to take into account the behavioral, cognitive, and emotional engagement in help instructors understand user engagement during the system development process.

However, the results of both multiple linear regression and classification analysis indicated that only the behavioral and cognitive components were significantly contributed to the predicting model. In other words, high frequencies of pausing, seeking, and stopping video were precisely predicted higher quiz scores. The results echo Li and Baker [34] study showing that cognitive engagement has its unique contribution in predicting academic achievement.

Because we did not find the strong contribution of single emotional engagement to predict students' quiz scores, the sequential analysis of the video watching behavior was performed. The results showed that the passed users tended to play a video through to the end, and the player stopped the video automatically. Contrary to the

passed users, the failed users tended to slow down or speed up a video through to the end, and then the player stopped the video automatically. This may be attributed to that emotional engagement should be further explored in more detail (e.g., classify the speed change event into speedup and slowdown). Therefore, we suggest that platform developers can design functions such as the ranking of video watching to encourage students to activate their video learning.

In the Phase 2, we 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.

In the Phase 3, this study used the click records of MOOCs videos. Firstly, the feature sequence of the viewing learning behavior is established by using the 4-gram approach, and the feature sequence was defined with the don't care mode as the type of learner's cognitive participation. Then, we used the K-Nearest Neighbor Classification (KNN) method, Support Vector Machines (SVM), and Artificial Neural Network (ANN) to predict whether students pass the course. Using the course data from two semesters, the predicted results of the built models were KNN accuracy 0.8624535, SVM accuracy 0.9442379, and ANN accuracy up to 0.955307263.

In addition, the weekly tutoring list of students was provided for teachers to supervise students' learning progress. There were 313 students who needed tutoring in the first week, 313 in the second week, 311 in the third week, 305 in the fourth week, and 297 in the fifth week.

Then, the prediction accuracy of the course data from the third semester was obtained, and the prediction accuracy of ANN under two hidden layers was as high as 0.986547085. We also used a second course to show that our approach can be

generalized for application to any new courses.

The prediction of learning outcomes can be presented through an analysis of learning records, course video clicking, and testing records. Due to the different natures of course videos, however, we have classified overall course videos into three types, i.e., theoretical, experimental, and analytic, and seven types of models were formed through combinations. There were 21 items of prediction accuracy of size 1~3 of ANN implemented based on the seven models. Items falling within the scope of ANN size < 4 and Accuracy > 0.9 were extracted. Using the simple accuracy verification, the study verified through video classification that the overall, theoretical, and experimental prediction accuracy values are close through matching. Therefore, the prediction effect can be achieved by using the clicking records of certain course videos (such as theoretical and experimental ones) instead of the whole data set.

Therefore, through the inference and prediction mechanism, this study analyzed the behavioral patterns and features of students' video browsing behaviors to determine the correlation between the video viewing behavior and learning outcomes, understand the features of students' learning behaviors with good or poor learning outcomes, and make predictions, which will provide a reference for teachers, so that teachers can implement tutoring measures in a timely fashion for students with poor learning outcomes and the course completion rate can be improved.

6.2 Conclusions

This study aimed to develop a software framework for MOOC learning analytics. The thesis has shown our approach is feasible and practical. As expected, we have completed the following results:

(1) After analyzing MOOC's log data of video viewing, we found that students with high scores tend to stop, pause, seek, and speed up/down video play at their own

pace. This may help teachers understand the behavior of user participation in the MOOC courses, including cognitive and emotional behaviors.

(2) We used engagement to predict student performance and performed a sequential analysis of video viewing behavior. The results indicated that the learners who pass the courses tend to play the video until the end. On the other hand, the failed ones tend to slow down or speed up the video until the end.

(3) Based on our MOOCs Learning Analytics Framework, the learning analysis service of MOOC is completed. We use the basic learning data course to develop feature sets and become core assets. These feature sets are used to organize relevant learning data to train various predictive models. Predictive models, data collection, data cleansing, and feature selection modules are saved as core assets.

(4) This framework provides a development environment of the software product line, which fully plays the role of reuse and makes the experiment achieve good results. Core assets and new requirements specifications can be easily used to quickly develop various applications to help learners that may not pass the course in the mid-term and any measures to reduce dropout rates.

(5) We used the data of a basic course being offered in two semesters to build a prediction model. Then the data from the third semester was applied to show the good prediction accuracy of the model. In addition, a second course was also used to show that our approach can be applied to any new course.

(6) To reduce the size of training data, we were able to divide the course videos into three types, namely theory, experiment and analysis in our experiments, and generate seven types of models with their combinations. With simple accuracy verification, predictive effects can be achieved by using click records from certain course videos (e.g., theoretical and experimental videos) rather than the entire data set.

6.3 Future Works

Although open online learning platforms are diverse, including a variety of different xAPI technologies and multimedia presentation modes, our MOOCs learning analytics framework can be used in future to reuse the core assets based on similar data records and do different product development more efficiently. Furthermore, it is beneficial to classify commonality and variability of framework components, which can become a part of the core assets to save more development time and cost.

Our future works include: (1) We plan to analyze different types of courses and those on the other MOOC platforms. Since the characteristics of course content and teaching objectives can be very different, we need to build more core assets and set up the environment for other MOOC platforms. (2) We will continue to analyze different types of courses and courses on other MOOC platforms. It is also useful to study the effect of improving the completion rate of the course. (3) We plan to establish platforms for video viewing feature generation and add various predictive models so that teachers of different courses can conduct learning analysis more easily. (4) This research result is developed by using the open source language of R. The prototype is expected to become open source, so that more interested teachers can participate in the research. (5) We will continue the study of learning sequence behavior and significance level of participation. (6) It is useful to detect learners' cheating behaviors.

References

[1] Project for the Promotion of MOOCs, A new e-learning of digital learning programs - 106-107 year-end final report, Ministry of Education, Taiwan, 2019.

[2] K. Judy, R. Peter, D. Elliot and K. Bob, "MOOCs: So Many Learners, So Much Potential," *Intelligent Systems*, pp. 70-77, May 2013.

[3] Software Engineering—Software Process and Software Process Models (Part 2) https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc

[4] S. Charles, "Teaching the World: Daphne Koller and Coursera," *Computer*, vol. 45, pp. 8-9, August 2012.

[5] L. B. 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.

[6] Y. Tabaa and A. Medouri, "LASyM: A Learning Analytics System for MOOCs," *Advanced Computer Science and Applications*, pp. 113-119, 2013.

[7] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z Gajos and R. C. Miller, "Understanding in-video dropouts and interaction peaks in online lecture videos," *Proceedings of the First ACM Conference on Learning @ Scale Conference*, Atlanta, Georgia, USA; pp. 31–40, 4–5 March 2014.

[8] C. Shi, S. Fu, Q. Chen and H. Qu, "VisMOOC: Visualizing Video Clickstream Data from Massive Open Online Courses," *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis) 2015*, Hangzhou, China; pp. 159–166, 14–17 April 2015.

[9] C. G. Brinton, S. Buccapatnam, M. Chiang, H. V. Poor, "Mining MOOC clickstreams: Video-watching behavior vs. in-video quiz performance," *IEEE*

Trans. Signal Process, 64, 3677–3692, 2016.

[10] L. Jiajun, L. Chao and Z. Li, "Machine Learning Application in MOOCs: Dropout Prediction," *Proceedings of the 11th International Conference on Computer Science & Education (ICCSE)*, Nagoya, Japan, pp. 52–57, 23–25 August 2016.

[11] X. Li, L. Xie and H. Wang, "Grade prediction in MOOCs," *Proceedings of the 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, Paris, France; pp 386–392, 24–26 August 2016.

[12] G. W. Dekker, M. Pechenizkiy and J. M. Vleeshouwers, "Predicting students drop out: A case study," *Proceedings of the 2nd International Conference on Educational Data Mining*, Cordoba, Spain; pp. 41–50, 1–3 July 2009.

[13] A. Elbadrawy, A. Polyzou, Z. Ren, M. Sweeney, G. Karypis and H. Rangwala, "Predicting Student Performance Using Personalized Analytics," *Computer*, 49, 61–69, 2016.

[14] C. G. Brinton and M. Chiang, "MOOC Performance Prediction via Clickstream Data and Social Learning Networks," *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, Kowloon, Hong Kong; pp. 2299–2307, 26 April–1 May 2015.

[15] K. Chorianopoulos, "A Taxonomy of Asynchronous Instructional Video Styles," *The International Review of Research in Open and Distributed Learning*, 19, 294–311, 2018.

[16] C. H. Yu, J. Wu, D. L Yang, M. C. Liu and A. C. Liu, "Video Watching Behavior Pattern Comparison of MOOCs Clickstream," *Proceedings of the Taiwan E-Learning Forum 2018*, Taipei, Taiwan, 23–25 May 2018.

[17] W. Greller and H. Drachsler, “Translating Learning into Numbers: A Generic Framework for Learning Analytics,” *Educational Technology & Society*, 15 (3), pp. 42–57, 2012.

[18] A. G. Picciano, “The Evolution of Big Data and Learning Analytics in American Higher Education,” *Journal of Asynchronous Learning Networks*, vol.16, no. 3, p9-20, Jun. 2012.

[19] G. Siemens and P. Long, “Penetrating the Fog: Analytics in Learning and Education,” *EDUCAUSE Review*, vol. 46, no. 5, p. 30, Jan. 2011.

[20] OpenEdu. Available online: https://www.openedu.tw/ (accessed on 2 February 2019).

[21] C. Romero, M.-I. López, J.-M. Luna and S. Ventura, “Predicting students’ final performance from participation in on-line discussion forums,” *Computers & Education*, vol. 68, pp. 458–472, 2013.

[22] Y. Meier, J. Xu, O. Atan and M. van der Schaar, “Predicting grades,” *IEEE Transactions on Signal Processing*, vol. 64, pp. 959–972, 2016.

[23] A. Anderson, D. Huttenlocher, J. Kleinberg and J. Leskovec, “Engaging with Massive Online Courses,” *Proceedings of the 23rd International Conference on World Wide Web 2014*, Seoul, Korea; pp. 687–698, 7–11 April 2014.

[24] F. Rebecca and C. Doug, “Examining Engagement: Analysing Learner Subpopulations in Massive Open Online Courses (MOOCs),” *Proceedings of the 5th International Learning Analytics and Knowledge Conference (LAK15)*, New York, NY, USA; pp. 1–8, 16–20 March 2015.

[25] K. Mohammad, and E. Martin, “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, Issue 1, pp 114–132, 2016.

[26] T. Sinha, P. Jermann, N. Li and P. Dillenbourg, “Your Click Decides Your Fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar; pp. 3–14, 25–29 October 2014.

[27] M. A. Chatti, A. L. Dyckhoff, U. Schroeder and H. Thüs, “A Reference Model for Learning Analytics,” *International Journal of Technology Enhanced Learning*, pp. 318-331, 2012.

[28] ISO/IEC TR 20748-1, Information technology — learning, education, and training — Learning Analytics Interoperability — Part 1: Reference model, 2016.

[29] M. A. Chatti, “The LaaN Theory. In: Personalization in Technology Enhanced Learning: A Social Software Perspective,” Aachen, Germany: Shaker Verlag, 2013, pp. 19-42. http://mohamedaminechatti.blogspot.de/2013/01/the-laan-theory.html

[30] R. N. Laveti, S. Kuppili, J. Ch, S. N. Pal and N. S. C. Babu, “Implementation of Learning Analytics Framework for MOOCs using State-of-the-art In-Memory Computing,” *2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH)*, DOI: 10.1109/ELELTECH.2017.8074997.

[31] N. H. Bakar, Z. M. Kasirun 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*, Volume 61, pp. 33-51, May 2015.

[32] P. Clement and L. Northrop, “A Framework for Software Product Line Practice, Version 5.0,” Software Engineering Institute,Carnegie Mellon University, December 2012, http://www.sei.cmu.edu/productlines/tools/framework/.

[33] D. Batory, R. Cardone and Y. Smaragdakis, “Object-Oriented Frameworks and Product Lines,” *Proceedings of the first conference on Software product lines*,

Pages 227-247, 2000.

[34] 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, pp. 325-346, September 2017.

[35] M. I. Hwang and R. G. Thorn, “The effect of user engagement on system success: A meta-analytical integration of research findings,” *Information & Management*, vol. 35, no. 4, pp. 229-236; DOI 10.1016/s0378-7206(98)00092-5, 1999.

[36] H. L. O'Brien and E. G. Toms, “What is user engagement? A conceptual framework for defining user engagement with technology,” *Journal of the American Society for Information Science and Technology*, vol. 59, no. 6, pp. 938-955; DOI 10.1002/asi.20801, 2008.

[37] K. Reed, “Users engage more with interface than materials at Welsh Newspapers Online Website,” *Evidence Based Library and Information Practice*, vol. 11, no. 3, pp. 91-92; DOI 10.1108/JD-10-2014-0149, 2016.

[38] L. Hakulinen, T. Auvinen and A. Korhonen, “The effect of achievement badges on students' behavior: An empirical study in a university-level computer science course,” *International Journal of Emerging Technologies in Learning*, vol. 10, no. 1, pp. 18-29; DOI 10.3991/ijet.v10i1.4221, 2015.

[39] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé and L. Getoor , “Learning latent engagement patterns of students in online courses,” *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence 2014*, 2014.

[40] J. A. Fredricks, P. C. Blumenfeld and A. H. Paris, “School engagement: Potential of the concept, state of the evidence,” *Review of Educational Research*, vol. 74, no. 1, pp. 59-109; DOI 10.3102/00346543074001059, 2004.

[41] Q. Li and R. Baker, “Understanding engagement in MOOCs,” *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 605-606, 2016.

[42] W. B. Cavnar and J. M. Trenkle, "N-Gram-Based Text Categorization," *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, USA; Volume 48113, pp. 161–175, 1994.

[43] A. Bailin and A. Grafstein, "The linguistic assumptions underlying readability formulae: A critique," *Language & Communication*, vol. 21, pp. 285–301, 2001.

[44] A. Kashyap and A. Nayak, "Different Machine Learning Models to predict dropouts in MOOCs," *In Proceedings of the 2018 International Conference on Advances in Computing*, Melbourne; pp. 80–85, Australia, 24–25 November 2018.

[45] A.-S. Raghad, H. Abir, L. Andy, K. Robert, L. Janet and R. Naeem, "Machine Learning Approaches to Predict Learning Outcomes in Massive Open Online Courses," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, pp. 713–720, 14–19 May 2017.

[46] 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*, 10: 1006-1010, 2010.

[47] B. Xu and D. Yang, "Motivation Classification and Grade Prediction for MOOCs Learners," *Computational Intelligence and Neuroscience*, doi:10.1155/2016/21746132016.

[48] S. Fauvel, and H. Yu, "A Survey on Artificial Intelligence and Data Mining for MOOCs," arXiv:1601.06862, 2016.

[49] T. Y. Yang, C. G. Brinton, J. W. Carlee and M. Chiang, "Behavior-Based Grade Prediction for MOOCs Via Time Series Neural Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11. doi:10.1109/JSTSP.2017.2772681, 2017.

[50] V. Trowler, "Student Engagement Literature Review," The Higher Education Academy 2010.

[51] J. Sinclair and S. Kalvala, “Student engagement in massive open online courses,” *International Journal of Learning Technology*, vol. 11, no. 3, pp. 218-237; DOI 10.1504/ijlt.2016.079035, 2016.

[52] R. Ferguson and D. Clow, “Examining engagement: analysing learner subpopulations in massive open online courses (MOOCs),” *Book Examining engagement: analysing learner subpopulations in massive open online courses (MOOCs)*, Series Examining engagement: analysing learner subpopulations in massive open online courses (MOOCs), ed., Editor ed.^eds., ACM, 2015, pp. 51-58.

[53] 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, no. 1, pp. 114-132; DOI 10.1007/s12528-016-9126-9, 2017.

[54] D. Benavides, S. Segura, A. Ruiz, “Automated analysis of feature models 20 years later: a literature review,”’ *Information Systems*, Volume 35, Issue 6, Pages 615-636, September 2010.

[55] D. Beuche and M. Dalgarno, “Software Product Line Engineering with Feature Models,” *Overload Journal*, Vol. 78, pp. 5-8, 2007.

[56] P. Clements and L. Northrop, “Software product lines - practices and patterns,” Addison Wesley, 2002.

[57] Chinese Open Education Consortum, URL: https://copeneduc.org/, April. 2017.

[58] C. H. Yu, J. Wu and A. C. Liu, “Predicting Learning Outcomes with MOOCs Clickstreams,” *In 2nd Eurasian Conference on Educational Innovation 2019*, Singapore, 27–29 January 2019.

[59] 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 No. 5, pp. 1175-1186, 2018.

[60] S. M. Wang, et al., "Analyzing the knowledge construction and cognitive patterns of blog-based instructional activities using four frequent interactive strategies (problem solving, peer assessment, role playing and peer tutoring): A preliminary study," *Educational Technology Research and Development*, vol. 65, no. 2, pp. 301-323; DOI 10.1007/s11423-016-9471-4, 2017.