

# **ANALYSING MOODLE LEARNING BEHAVIOUR ABOUT VIRTUAL PATIENTS**

**Menglin Xu**

University of Tampere  
Faculty of Natural Sciences  
Degree Programme in Computer Sciences  
M.Sc. Thesis  
Supervisor: Kati Iltanen  
June 2017

University of Tampere  
Faculty of Natural Sciences  
Degree Programme in Computer Sciences  
MENGLIN XU: Analysing Moodle Learning Behaviour about Virtual Patients  
M.Sc. Thesis, 48 pages, 23 appendix pages  
June 2017

---

With the development of Internet, online learning management systems (LMSs) have been used widely for providing teaching platforms. The vast quantities of data that LMSs generate daily are difficult to manage manually. Thus, educational data mining (EDM) is applied to solve this problem.

In this thesis, EDM is applied on Moodle log data of a medical course. This course was arranged by problem-based learning (PBL) method, which uses virtual patients (VPs) as a problem, to improve students' diagnostic skills. The aim of this thesis is to analyse Moodle learning behaviour related to the usage of VPs and implement a set of Python algorithms to handle such kind of data. There are two ways are utilised to analyse Moodle log data by EDM: applying data mining techniques and implementing Python scripts. The techniques applied on the first way are attribute weighting and generalized sequential patterns (GSP), while the second way provides Python algorithms about extracting frequencies, sessions, and relationship tables.

This thesis shows learning behaviour records and patterns about the usage of each VP. In addition, it gives information about the overall usage of different kinds of activities and resources that Moodle offers. Moreover, Python algorithms implemented in this thesis provide tools to extract frequencies, sessions, and relationship tables of Moodle log data for further research.

**Keywords:** Educational Data Mining, Problem-based Learning, Sequential Pattern Mining, Moodle, Learning Pattern, Learning Management System.

# CONTENTS

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>BACKGROUND</b>	<b>4</b>
2.1	Problem-based Learning . . . . .	4
2.2	Educational Data Mining . . . . .	6
2.3	Previous work . . . . .	9
<b>3</b>	<b>DATA</b>	<b>10</b>
<b>4</b>	<b>APPLYING DATA MINING TECHNIQUES</b>	<b>12</b>
4.1	Procedures and Instruments . . . . .	12
4.2	Ranking Moodle Activities . . . . .	14
4.3	Applying Sequential Pattern Mining Techniques . . . . .	18
<b>5</b>	<b>IMPLEMENTING PYTHON METHODS</b>	<b>22</b>
5.1	Extracting Frequencies . . . . .	23
5.2	Extracting Sessions . . . . .	25
5.3	Extracting Relationships . . . . .	27
<b>6</b>	<b>RESEARCH RESULTS</b>	<b>33</b>
6.1	Attribute Weighting Results . . . . .	33
6.2	Summary Table Results . . . . .	35
6.3	Generalized Sequential Patterns Results . . . . .	38
6.4	Python Method Results . . . . .	41
<b>7</b>	<b>CONCLUSION</b>	<b>47</b>
	<b>REFERENCES</b>	<b>49</b>
	<b>APPENDICES</b>	

## ABBREVIATIONS AND SYMBOLS

<b>GSP</b>	Generalized Sequential Patterns
<b>VP</b>	Virtual Patients
<b>SPADE</b>	Sequential Pattern Discovery using Equivalent Classes
<b>PBL</b>	Problem-based Learning
<b>IVIMEDS</b>	International Virtual Medical School
<b>EDM</b>	Educational Data Mining
<b>LMS</b>	Learning Management System
<b>Moodle</b>	Modular Object-oriented Developmental Learning Environment
<b>SDL</b>	Self-directed Learning
<b>IP</b>	Internet Protocol

# 1 INTRODUCTION

With the development of Internet, online learning has become widespread in various countries and universities. Typically, universities and other education institutions use learning management systems (LMSs) as instruction, communication, and reporting tools for practical educational purposes [40]. The use of Internet in education has created e-learning (also referred to as Web-based education), a context in which increasing amounts of teaching-learning interactions information are endlessly generated and ubiquitously available [13]. Comparing with traditional offline education that transmits skill and knowledge in classroom environments, e-learning and learning management system have been adopted for knowledge sharing and course material delivery by providing online instruction, communication, collaboration and reporting tools [33, 43].

Educational data mining (EDM) is a multidisciplinary area which applies various data mining techniques into educational data to better understand behaviour of learners [8]. The vast quantities of data that LMSs generate are difficult to manage manually [32]. Meanwhile, there are not specific tools by which teachers can assess all online activities in order to evaluate the contents of this course and its effectiveness for learning process [44]. Thus, the use of data mining in educational area can solve this problem [43].

Moodle (Modular Object-Oriented Developmental Learning Environment), one of the popular open-source LMS, is used to help teachers to create effective online learning environments and communities [34]. It has been used at institutions and schools all over the world. The Moodle log data as educational data, which contains information of students' learning behaviour, is the precious resources for researchers to analyse and utilize [12].

In this thesis, educational data mining is applied to research Moodle log data from a medical course arranged at the University of Tampere. The Moodle log data contains information of the usage of Virtual Patients (VPs) and the contents of VP are from International Virtual Medical School's (IVIMEDS) [21] virtual patient applications [40]. Each VP has virtual personal information and detailed symptom description. VP as a problem in learning process is a good experimental object to research how students follow problem-based learning method [10].

Problem-based learning (PBL) is an instructional method in which students learn through problem solving and use specific problems to improve skills and knowledge. Given a problem case or scenario, a cyclic process is used by students to acquire knowledge independently or in group in order to obtain the learning outcomes. In PBL, students focus

on a complex problem that does not have a single solution. They work in collaborative groups to figure out what they should learn in order to solve such a problem. Hmelo-Silver [18] proposes five goals of PBL which are: flexible knowledge, effective problem-solving skills, self-directed learning (SDL) skills, effective collaboration skills, and intrinsic motivation. [18, 40, 41]

In this thesis, Moodle log data for a medical course will be used as a case study and two ways are utilised to analyse data by EDM. One is to apply data mining algorithms to analyse Moodle logs; the other is to implement Python scripts to obtain results on the usage of Moodle activities and resources. In the first way to launch EDM, the algorithms applied on the data mining are attribute weighting and sequential pattern mining techniques. Because some students usually follow the same pattern to read and view the material on Moodle, it is proposed that there is a relationship between the Moodle learning behaviour, the order of events viewed by the students and following problem-based learning approach.

The overall aim of this thesis is to analyse Moodle learning behaviour about Virtual Patients, which are used as online teaching materials in a medical course at the University of Tampere. The specific aims are as follows:

- 1 To gain some useful online learning patterns from Moodle logs about the usage of Virtual Patients. Learning patterns are useful for students to reflect on their own learning habits by which good suggestions can be given to improve students' learning skills.
- 2 To provide teachers with useful information to improve or enrich the content or form of virtual patterns, to monitor and give prerequisites information. As a result, teachers may possibly better arrange courses based on PBL method.
- 3 To implement new tools to handle the Moodle logs by Python language. Python scripts provide algorithmic structures to extract frequency tables, session tables, and relationship tables from Moodle logs.

The rest of the thesis is organized as follows. Chapter 2 mainly focuses on literature related to the topics of problem-based learning and educational data mining. Chapter 3 gives a description of Moodle log data which is applied in the process of data mining as the original data set. In Chapter 4, two data mining techniques: attribute weighting and sequential pattern mining are applied on RapidMiner tool to discover learning patterns for further evaluation. Meanwhile, the procedure of how to use Python functions to generate

frequencies, sessions, and relationship tables is introduced in Chapter 5. Finally, Chapter 6 summarizes the results as well as the final conclusion of the thesis work and future research are explained in Chapter 7.

## 2 BACKGROUND

This chapter introduces the necessary background information of problem-based learning (PBL) and educational data mining (EDM). Section 2.1 discusses the concept and theoretical basis of PBL and also introduces Virtual Patient (VP) as a sample problem applied in PBL method in medical education. Section 2.2 defines EDM and EDM stakeholders as well as introduces approaches to execute EDM. Moreover, two trains of thought to analyse Moodle log data, using data mining techniques and implementing methods by Python, are proposed from the theoretical basis of EDM.

### 2.1 Problem-based Learning

For over twenty years, problem-based learning (PBL) has been widely used in many countries across different fields of education including medicine, economics, mathematics, business studies, engineering, architecture, and law [31]. It is a small-group teaching method in which students learn both knowledge and train of thought through problem solving process [35]. Poikela and Nummenmaa [31] proposed that learning at work is the theoretical basis of problem-based learning, which can show its importance of practicality and authenticity.

In many medical schools, PBL has supplanted the traditional lecture-based learning [7]. Lecture-based learning is usually utilized during formal schooling and training. With the gap between school education and actual work becoming increasingly wide, simply lecture-based learning can not satisfy students' requirements of skills and knowledge needed in working life. Working life requires skills including problem solving, cooperation, continuing learning and independent knowledge acquisition [31]. To a certain extent, the PBL method uses appropriate problems to extend knowledge and deepen understanding.

Dolmans et al. [14] indicate three essential characteristics of PBL: problems as a stimulus for learning, tutors as facilitators, and group work as stimulus for interaction of PBL [26]. Although the process of PBL tutorials can vary according to the situation, the Maastricht seven steps method is usually utilized as a guide for organizing a PBL tutorial [2, 41]:



- 1 Identify the terms and objects in this tutorial.
- 2 Define the problems to be discussed.
- 3 Discuss the problems on a brainstorming session.
- 4 Review discussion process and arrange explanations into tentative solutions.
- 5 Formulate learning objectives.
- 6 Self-study information related to each learning objective.
- 7 Share results of private study in group and tutor assesses the performance.

VPs are good examples of how to apply PBL in medical education. They are high quality learning resources used in medical education and each VP contains personal information and detailed symptom description. The VPs used in this research come from the International Virtual Medical School (IVIMEDS), which is a global collaboration of medical educators dedicated to improving the learning experiences of medical students by the appropriate deployment of technology in support of learning and teaching [21].

In the real work life of medical students, the most important task is interviewing patients and making a preliminary diagnosis according to patients' condition description. Virtual Patients include patients' information that can provide a more realistic experience than a lecture. As mentioned before, practicality and authenticity are significant for the PBL method. The practicality of the PBL method is well proven by students personally simulating the diagnostic process on virtual patients. In addition, information of Virtual Patients, which are extracted from real patients' information for medical student to deal with, shows the authenticity of the PBL method.

## 2.2 Educational Data Mining

Educational data mining (EDM) is a mixed subject which applies data mining, machine-learning, and statistical methods into various kinds of educational data [33]. EDM is concerned with developing methods for discovering different kinds of patterns and models from educational area, then utilizing those mined results to better understand students and the mode they learn [8]. The main objective of EDM in this thesis is to obtain online learning patterns about the usage of VPs by analysing Moodle log data.

Although an intuitive thought seems to include only two main groups, the learners and the educators as stakeholders in EDM, there are actually more groups which should be involved. Different groups look at educational information from different angles, according to their own objectives and reasons for participating in EDM. All the stakeholders and their objectives for participating in EDM can be seen in Table 2.1. [17, 33]

**Table 2.1.** EDM Stakeholder [33].

<b>Stakeholder</b>	<b>Objectives for Participating in EDM</b>
Learners/ Students	E.g. to further improve learning skills by resources and learning tasks; to suggest interesting learning experiences to follow for students; to personalize e-learning.
Educators/ Teachers/ Instructors	E.g. to get feedback from learners; to analyse students' learning behaviour; to predict student performance; to group students; to determine more effective activities.
Course Developers/ Educational Researchers	E.g. to maintain course; to evaluate the structure of content and its effectiveness; to compare data mining techniques in order to optimize recommend method.
Organizations/ Universities/ Training Companies	E.g. to enhance the decision processes in learning institutions; to suggest certain valuable course for each class of learners; to find the most cost-effective way of improving learning outcomes; to select the most qualified applicants for graduation.
Administrators/ Network Administrators/ System Administrators	E.g. to develop the best way to organize institutional resources; to evaluate teachers and curricula; to utilize available resources more effectively.

Romero et al. [34] provide four steps of educational data mining process: collect data, preprocess data, apply data mining, and evaluate results. Moodle stores the interaction and usage information of students in this thesis. Before executing data mining, there is a step that the collected data needs to be cleaned and prepared into an appropriate format, which called preprocessing. Then, the data mining algorithms are applied on the prepared data to discover the interesting patterns and models as well as obtain knowledge from the data. Finally, the obtained results should be evaluated by the researchers for further actions. A whole EDM process consists of these four steps.

Sukhija et al. [38] propose that many EDM tasks are mapped to conventional data mining techniques. Because there are many different kinds of techniques or methods popular in EDM, Baker et al. [8] categorize five categories: prediction, clustering, discovery with models, relationship mining, and extracting data for human judgment. Table 2.2 shows these five types of techniques with their concepts and corresponding purposes in EDM. In this thesis, there are two different trains of thought to execute EDM in Moodle logs. The first one is from relationship mining and prediction approach mentioned above, which uses data mining algorithms in Rapidminer to obtain learning patterns. The second way to execute EDM is to extract data for human judgment. In the second way, Python scripts are implemented to obtain the desired results in order to analyse students' behaviours.

Baker et al. [8] propose four main applications that have received particular attention within EDM: improving student models which provide students' characteristics; learning the pedagogical support from educational software; discovering models of EDM's knowledge structure; and scientific discovery of learners and learning. Wu and Leung [42] propose the statistical software, e.g., SPSS, can be used with educational data as well as summaries and reports about students' behaviour can be provided by the statistical analysis. Association rules, prediction, clustering, and sequential pattern analysis have been used to discover the way of enhancing the web-based learning interest of students [43]. Especially, Baroque et al. [11] analyse the students' Moodle log to enhance the Moodle course contents and online learning environment and Myszkowski et al. [29] use e-learning data to improve the effectiveness of learning process. Meanwhile, Dong and Pei [15] propose the sequential pattern mining aims to discover the relationships among sequential events to obtain learning patterns. The mined data and knowledge enhance the learning processes and also provide an improved insight [38]. All of these state of art studies in EDM provide this research a good theoretical basis.

**Table 2.2.** Five kinds of techniques in EDM.

<b>Technique</b>	<b>Concept</b>	<b>Purpose in EDM</b>
Clustering	Clustering is an unsupervised learning technique which aims to discover similarities among data and group similar data according to their characteristics.	Grouping similar students based on learning behaviour. Discovering the learning patterns of new students.
Relationship mining	Relationship mining discovers relationships between variables in a data set. There are four types of relationship mining, namely association rule mining, correlation mining, causal data mining, and sequential pattern mining [8].	Discovering associations of curricular or learning behaviour.
Prediction	Prediction develops a model to estimate the value of the variable which is not known [22]. There are three types of prediction, namely regression, classification, and density estimation [8].	Predicting whether a student will pass a course or not.
Discovery with models	In the discovery with models process, a model is developed by clustering, prediction etc. techniques, then the model is used to further analysis.	Discovering the relationship among different learning behaviour of students. Analysing student characteristics.
Extracting data for human judgment	Human beings sometimes make inferences about data as a supplement for data mining methods.	Labelling data to help to improve the prediction model. Discovering learning behaviour and patterns of students.

## 2.3 Previous work

Walldén and Mäkinen [40] reveal the quantity of students who view the materials and the duration of each activity. They classify students into two groups according to whether the students follow PBL and present the result of different groups. This result shows that most of the PBL followers are also active Moodle users who have no less than 124 overall Moodle sessions (124 is the median of the numbers of the students' Moodle sessions during this course). They propose that the usage of VPs should have relationships with the rhythm of the tutor sessions and lectures rather than cause a peak just before final exam. This would indicate that the PBL method has satisfactory performance in the course. Furthermore, sequential learning pattern as a new analysing point and discovering patterns about how students check certain materials related to the description of the VP's diagnosis are proposed in the further work part. In this thesis, generalized sequential patterns (GSP) algorithm will be used as a data mining technique to obtain students' sequential learning patterns. In addition, the Python methods are implemented to obtain interesting learning patterns as a supplement.

### 3 DATA

Moodle records detailed logs of all students' activities which include important information. For example, this can be useful to check whether everyone has spent the required online time to do a certain task [34]. Moodle stores the logs in a relational database, such as PostgreSQL or MySQL. In this research, the original data is a CSV version of Moodle logs extracted from PostgreSQL. A fragment of original data is shown in Figure 3.1. The variables and their meanings in the original Moodle log data are summarized and shown in Table 3.1.

course	ip_address	user_id	starting_time	next_starting_time	duration	duration	event	information	row_id
III vsk Infektiojakso	87.93.114.74	101	20.12.2010 20:03	20.12.2010 20:03	00:00:00	0	course view	III vsk Infektiojakso	10
III vsk Infektiojakso	87.93.114.74	101	20.12.2010 20:03	20.12.2010 20:06	00:03:00	3	resource view	Palautelomake	11
III vsk Infektiojakso	193.167.131.1	181	11.1.2011 16:51	1.12.2011 13:29	323 days 20:38:4	466358	resource view	Lasten septiset infektio	295
III vsk Infektiojakso	91.155.228.140	254	11.1.2011 17:59	\N	\N	\N	course view	III vsk Infektiojakso	296
III vsk Infektiojakso	91.152.3.148	315	6.4.2011 8:08	6.4.2011 8:08	00:00:00	0	course view	III vsk Infektiojakso	797
III vsk Infektiojakso	91.152.3.148	315	6.4.2011 8:08	6.4.2011 8:08	00:00:00	0	resource view	Virtual patient Terttu S	798
III vsk Infektiojakso	91.152.3.148	315	6.4.2011 8:08	6.4.2011 8:09	00:01:00	1	course view	III vsk Infektiojakso	799
III vsk Infektiojakso	91.152.3.148	315	6.4.2011 8:09	6.4.2011 8:09	00:00:00	0	course update mod	resource 12527	800
III vsk Infektiojakso	91.152.3.148	315	6.4.2011 8:09	6.4.2011 8:10	00:01:00	1	resource update	Virtual patient Terttu S	801

**Figure 3.1.** A fragment of Moodle log data.

**Table 3.1.** Variables and labels of Moodle logs.

Variable	Label
course	name of the course
ip_address	ip address of Moodle users
user_id	identify each Moodle user
starting_time	starting time of an event
next_starting_time	starting time of the next event
duration	duration of an event(next_starting_time minus previous starting_time)
duration_minutes	transfer duration of an event in minutes
event	event in Moodle
information	some items recorded as Virtual Patients and other materials
row_id	the row number in the original file

In Table 3.1, there are ten variables in the original data source, namely course, ip\_address, user\_id, starting\_time, next\_starting\_time, duration, duration\_minutes, event, information and row\_id. Course variable is used to indicate in which course each log item has taken place. In this thesis, one infectious disease course of the University of Tampere called "III vsk Infektiojakso" is investigated and researched, which means course variable can be seen as constant in this research. IP (Internet Protocol) address as a location

signal recorded in CSV log file can be utilized to analyse whether learning place has an influence on final marks, according to campus IP address or non-campus IP address. User id is a significant variable that can be used to identify each Moodle user. Every user on Moodle has their own id number; and in this course, there are 329 students, which means the user\_id varies from 1 to 329 in the original CSV log file. Next, there are four variables about time: starting\_time, next\_starting\_time, duration and duration\_minutes. Here the duration and duration\_minutes of an event are calculated from previous two variables. Moodle does not keep the duration of each event in the log files, only the start time of each event is recorded. With the information of the next event's start time (next\_starting\_time), the duration of certain event can be calculated as the interval of the next event's start time (next\_starting\_time) and certain event's start time (starting\_time). Then, duration\_minutes transfers the duration of an event in minutes in order to become a quantitative attribute that can be handled by some learning schemes and algorithms in data mining tool RapidMiner. Furthermore, event and information are two important variables in which a certain event and its detailed description are recorded. Specifically, the type of one event are recorded as the content of event variable, e.g., resource view, course view, course enrol, forum view discussion, and discussion mark read. In this thesis, the mainly focused type of event is 'resource view'. Moreover, the title of resource will be recorded as the content of information variable. Finally, the row\_id as a variable is used to identify the row numbers of each log items in order.

This research investigates the use of Moodle when students follow the problem-based learning (PBL) method. In addition, VPs are used as main resources to be viewed and learned in the process of PBL. Some of information variables of VPs with detailed identifiers are listed in each data set, meanwhile, the content of event variable is usually shown as 'resource view'.

However, Moodle log data has several disadvantages. Moodle just records the start time of every event. Thus, the duration is calculated as the interval of two events instead of an accurate duration of a certain event. In addition, Moodle does not keep the time stamps in the level of seconds, thus, it is a problem to identify the order of events that happened at the same minute.

## 4 APPLYING DATA MINING TECHNIQUES

Mazza and Dimitrova [28] propose that tabular tracking data is usually incomprehensible with a poor logical organization. Thus, steps to prepare data for mining is necessary. After data preparation, the corresponding data set has been generated as data source which can be applied into the next experimental steps. Section 4.1 introduces the procedures of this research method and the RapidMiner is given a full description as research instrument. Sections 4.2 and 4.3 introduce the algorithms applied on RapidMiner tool, which are attribute weighting (Weighting by Information Gain, and Relief) and sequential pattern mining (Generalized Sequential Patterns).

### 4.1 Procedures and Instruments

The research method that apply data mining techniques has four steps as follows:

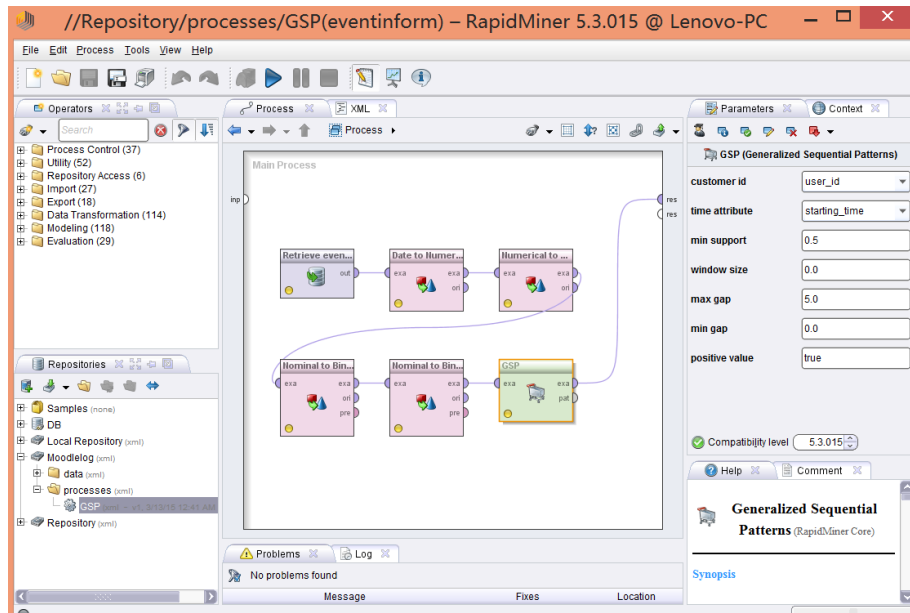
- 1 Preparing data set through Moodle logs
- 2 Selecting data mining algorithms
- 3 Performing data mining experiments
- 4 Analysing experimental results.

Although Moodle offers tabular tracking data on students' activities, it is not flexible enough to satisfy the researchers' needs for evaluating students' interactions on the platform [28]. Therefore, it is necessary to preprocess the data. The goal of the research method was to understand Moodle log data and apply data mining algorithms to discover learning patterns for further evaluation. Because the data source is based on log information of a medical course, knowledge about how the students learn on Moodle system can be acquired by the analysis and evaluation of experimental results.

For the past twenty years, free software tools for data mining have been in development to offer all the researchers publicly available data analysis platforms. RapidMiner is a general data mining tool, for which previous version 5 or lower are open source. This research uses RapidMiner tool which provides an user-friendly integrated environment for machine learning, data mining and other functions to obtain the learning patterns. This tool has the availability and maturity of a large number of data mining algorithm



implementations as well as supports all steps of the data mining process including results visualization, validation, and optimization. Figure 4.1 shows the interface of RapidMiner. [23]



**Figure 4.1.** The interface of RapidMiner.

## 4.2 Ranking Moodle Activities

Hunt et al. [19] proposed information gain as a measure for ranking attributes. Later, many researchers have used this notion of feature selection to estimate the attributes' quality. Information gain increases with the average purity of the subsets which an attribute produces. Thus, the strategy for attribute weighting by information gain is ranking attributes that results in greatest information gain. [19]

Han et al. [16] summarize the basic conception related to information gain, such as entropy, information, and information gain. Entropy is a measure of uncertainty associated with a random variable [9]. Given a probability distribution, the information required to predict an event is the entropy of the distribution [39]. The higher the entropy, the higher the uncertainty. The calculation method for a discrete random variable  $X$  taking  $n$  distinct values  $\{X_1, \dots, X_n\}$  and let  $p_n$  be the probability that  $X_i$  belongs to  $X$  is the following:

$$entropy(X) = entropy(p_1, p_2, \dots, p_n) = -p_1 \log_2(p_1) - p_2 \log_2(p_2) \dots - p_n \log_2(p_n). \quad (4.1)$$

For example, Table 4.1 lists four weather attributes which might influence the result of playing or not playing. Among these items, there are nine positive results while five negative results. The information (entropy) needed to classify a tuple in attribute 'play or not', which is expressed as follows:

$$\begin{aligned} info(play?) &= I([9, 5]) \\ &= entropy\left(\frac{9}{14}, \frac{5}{14}\right) \\ &= -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.410 + 0.530 = 0.940. \end{aligned} \quad (4.2)$$

Information gain is one of attribute selection measures which selecting the attribute with the highest information gain. Information gained by branching on attribute is:

$$gain() = info_{beforeSplit}() - info_{afterSplit}(). \quad (4.3)$$

Continuing the example of Table 4.1, there are three values of attribute 'outlook': sunny, overcast, and rainy. Let partition the results into subsets by this attribute as Figure 4.2. The expected information ( $Info_{outlook}(play?)$ ) based on the partitioning into subsets by

**Table 4.1.** Weather attributes which might influence the result of playing or not playing [16].

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rain	mild	high	false	yes
rain	cool	normal	false	yes
rain	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rain	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rain	mild	high	true	no

attribute 'outlook' is calculated as:

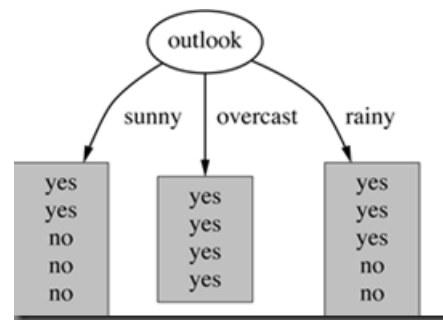
$$\begin{aligned}
info_{outlook}(play?) &= I([2, 3], [4, 0], [3, 2]) \\
&= \frac{5}{14} \times I([2, 3]) + \frac{4}{14} \times I([4, 0]) + \frac{5}{14} \times I([3, 2]) \\
&= 0.347 + 0 + 0.347 \\
&= 0.694bits.
\end{aligned} \tag{4.4}$$

Hence, the information gain of attribute 'outlook' is calculated as:

$$\begin{aligned}
gain(outlook) &= info_{beforeSplit}() - info_{afterSplit}() \\
&= info(play?) - info_{outlook}(play?) \\
&= 0.940 - 0.694 = 0.246.
\end{aligned} \tag{4.5}$$

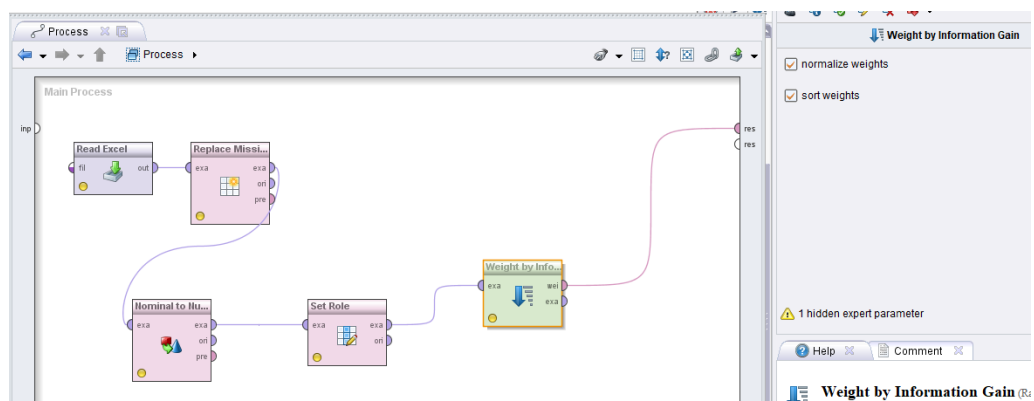
After the calculation of information gain of each attribute, the weighting by information gain algorithm ranks all attributes by value of information gain from the greater to the smaller.

Kira and Rendell [24] propose an efficient attribute weighting algorithm called Relief. Its key idea is to estimate attributes according to how well the values of attributes distinguished by other near instances. Relief algorithm bases on the collection of all the features related to statistics and it is also fairly noise-tolerant, which are important for real world domains. Relief is also applied to rank Moodle events. [24, 25]

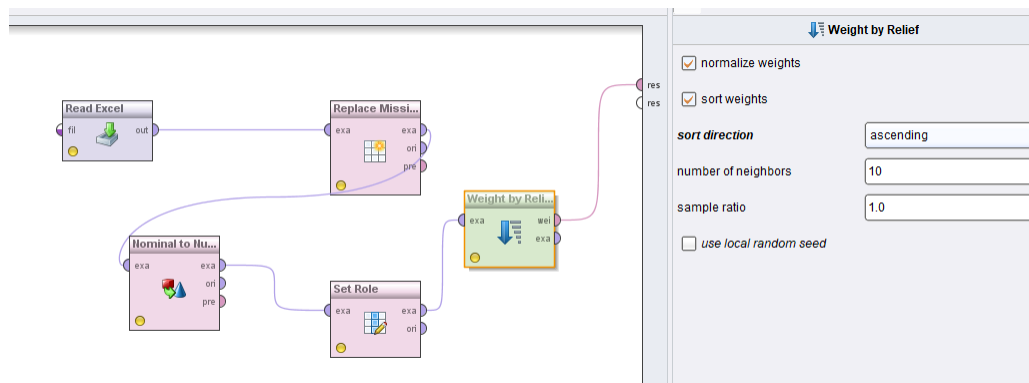


**Figure 4.2.** Partition by attribute outlook [16].

When using RapidMiner to execute these two attribute weighting algorithms, the nominal to numerical operator is used for changing the type of non-numeric variable 'event' to a numeric type [6]. Detailed explanation of this variable is introduced in Chapter 3. This operator also maps all values of variable 'event' to numeric values as ranked attributes [6]. Then, the variable 'duration\_minutes' is setted up as 'label' role which means the ranking of the attributes are gained by how much the attributes contribute to the 'label' role. Here the value of variable 'duration\_minutes' is separated into two classes 'Yes' and 'No' by setting a threshold. This thesis uses 10 as the value of the threshold because 10 minutes are the minimum reasonable time interval for an effective Moodle event. The value of variable 'duration\_minutes' that no less than 10 is setted as 'Yes', otherwise the value is setted as 'No'. Finally, these two attribute weight operators calculate the weight of attributes by information gains and Relief, respectively. Figures 4.3 and 4.4 illustrate the processes of executing weighting by information gain and Relief algorithms on RapidMiner tool.



**Figure 4.3.** The interface of RapidMiner about executing Weight by Information Gain algorithm.



**Figure 4.4.** The interface of RapidMiner about executing Weight by Relief algorithm.

### 4.3 Applying Sequential Pattern Mining Techniques

Pattern is a valid, understandable, novel, and potentially useful local feature of data, which is a kind of interesting information. Frequent pattern is a pattern or a set of items that appears frequently in a data set and it is a foundation for association analysis, correlation analysis, sequential patterns etc. Han et al. [16] introduce some basic concepts of frequent patterns, e.g., itemset, support, and confidence. Itemset is a set of one or more items, while k-itemset X taking k distinct values  $\{x_1, \dots, x_k\}$ . There are two kinds of support: absolute support, and relative support. The absolute support of X means the occurrence or frequency of an itemset X. The relative support is the fraction of transactions that contains X, which is the probability that a transaction contains X. A support is a measure for the interestingness of an itemset and we claim an itemset X is frequent if X's support is no less than a min support threshold. Association rules find all the rules  $X \rightarrow Y$  with minimum support and confidence. Support is the probability that a transaction contains union of X and Y. Confidence is the conditional probability that a transaction having X also contains Y. [3, 16]

Table 4.2 gives a supermarket example of data set with transaction id and items bought at each transaction. If the min support and min confidence are both setted as 50%, the frequent patterns are: Beer:3, Nuts:3, Pizza:4, Eggs:3, {Beer,Pizza}:3. Meanwhile, the association rules will be Beer  $\rightarrow$  Pizza (60%, 100%), Pizza  $\rightarrow$  Beer (60%, 75%), etc.

**Table 4.2.** Example of a data set [16].

Transaction Id	Items Bought
10	Beer, Nuts, Pizza
20	Beer, Coffee, Pizza
30	Beer, Pizza, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Pizza, Eggs, Milk

Support and confidence are not good to indicate correlations [4]. For example, the constraints of a strong rule can be: min\_support  $\geq$  3% and min\_confidence  $\geq$  60%. By contrast, lift is a measure of correlated events, which is used for indicating correlations in interestingness measure. It is defined as:

$$lift = \frac{P(X \cup Y)}{P(X)P(Y)}. \quad (4.6)$$

Sequential pattern mining finds all frequent subsequences whose occurrence frequency in the set of sequences is no less than min support [20]. A sequence  $S$  ( $S = \langle e_1, e_2, e_3, \dots \rangle$ ) is an ordered list of transactions associated with a single data object, among which the ordinal relations are based on temporal or spatial precedence. Each transaction  $e_i$  ( $e_i = (i_1, i_2, \dots, i_k)$ ) contains a set of items that are unordered in a transaction. An item can appear at most once in a transaction of a sequence but multiple times in different transactions [4]. [1]

Nguyen and Do [30] propose two kinds of different approaches of sequential pattern mining. The first one is a candidate generation approach which includes AprioriAll, Generalized Sequential Patterns (GSP) and SPADE (Sequential Pattern Discovery using Equivalent classes), while the second approach is a pattern growth approach. This research applies GSP algorithm in Rapidminer to discover the sequential patterns.

Generalized Sequential Patterns is a popular algorithm used for finding generalized sequential patterns [37]. There are two steps to conduct GSP: generating all the 1-item frequent sequences; and repeating candidate generation, candidate pruning, support counting, and candidate elimination processes until no new frequent sequences are found. Candidate generation merges pairs of frequent  $(k-1)$ -subsequences found in the  $(k-1)_{th}$  pass to generate candidate sequences that contains  $k$  items. Candidate pruning prunes candidate  $k$ -sequences that contain infrequent  $(k-1)$ -subsequences. Support counting makes a new pass over the sequence database to find the support for these candidate sequences. Candidate elimination eliminates candidate  $k$ -sequences whose actual support is less than min support. [36]

The input transaction data of GSP is shown in Figure 4.5. Algorithm 1 is used to generate the transaction data. The algorithm is used for extracting sessions in certain duration for each user with timestamp. It will read one row from input file every time and check whether the user id in this row is already in the result dictionary. If the user id exists in the result dictionary's keys, it will add the VP into an existed session or generate a new session with this timestamp according to the time difference between this row's starting time and last row's starting time, and put the session with timestamp into result dictionary. If not, it will generate a new session with timestamp and put it into result dictionary. Meanwhile, the VPs in rows will be added into a set called 'vp\_set', which is used for helping output CSV files of specified format. In this script, the timestamp is a UNIX timestamp.

id	timestamp	VP Petteri	V VP Merja	P VP Terttu	S VP Juhani	M VP Pekka	Oi VP Jennifer	VP Sirkka	S VP Carla	M VP Kerttu	V VP George	F VP Bertta	Berl VP Matti	N
4	57702120	0	0	0	0	0	0	0	0	0	0	0	0	0
4	60443640	0	0	0	0	0	0	0	0	1	0	0	0	0
5	59587740	0	0	0	0	0	0	0	0	0	0	0	0	0
5	60116880	0	0	0	0	0	0	0	0	0	0	0	0	0
9	58387320	0	0	0	0	0	0	0	0	0	0	0	0	0
14	57501900	0	0	1	0	0	0	0	0	0	0	0	0	0
14	59140560	1	0	0	0	0	0	0	0	0	0	0	0	1
14	59329680	0	1	0	0	0	0	0	0	0	0	1	0	0
14	59581680	0	0	0	0	0	0	0	0	0	0	0	0	0
14	60170640	0	0	0	0	0	0	0	0	0	0	0	0	0

**Figure 4.5.** A fragment of input data for GSP.

---

**Algorithm 1** Extract sessions that are in certain duration for each user with timestamp

---

**Input:**  $file\_path$ , the path of input file;  $duration$ , the duration of each session in seconds

**Output:**  $res$ , a dictionary of all qualified sessions and their users;  $vp\_sett$ , a set of all VPs

```

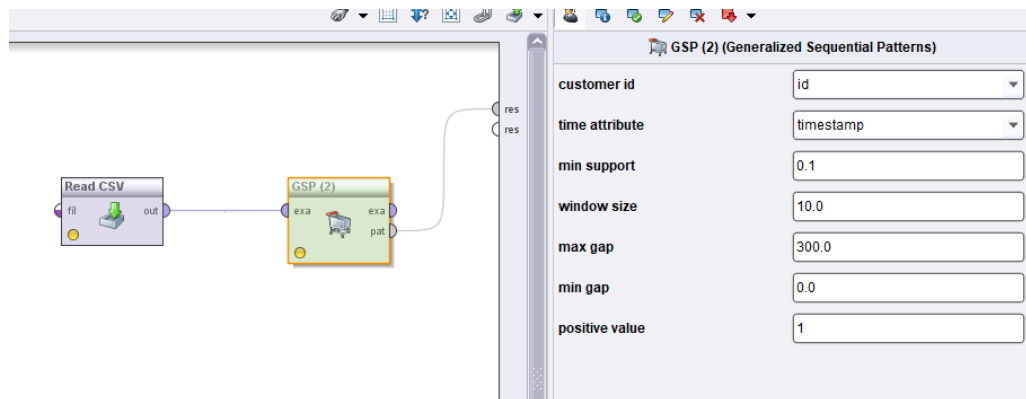
1: function  $get\_sessions\_in\_certain\_duration\_with\_timestamp(file\_path, duration)$ 
2:    $res \leftarrow$  empty dictionary //Initialize the variable for output
3:    $f \leftarrow open\_file(file\_path)$  //Open input file
4:    $vp\_set \leftarrow$  empty set //VP set for help
5:   for each  $row$  in  $f$  do //Read each row from input file
6:      $user\_id \leftarrow$  user id from  $row$ 
7:      $vp \leftarrow$  VP from  $row$ 
8:     add  $vp$  to  $vp\_set$ 
9:      $starting\_time \leftarrow$  starting time from  $row$ 
10:    if  $row$  is the header row then //Deal with the header row
11:      continue
12:    end if
13:     $timestamp \leftarrow$  UNIX timestamp for  $starting\_time$ 
14:    if  $user\_id$  is already in  $res$  then
15:      if the starting time of this  $row$  - the starting time of last  $row \leq duration$ 
16:        then
17:          append  $vp$  into last position's session in  $res[user\_id]$  //aka
18:           $res[user\_id][last\_one][1]$ 
19:        else
20:          append  $(timestamp, [vp])$  into  $res[user\_id]$ 
21:        end if
22:      else
23:         $res[user\_id] \leftarrow [(timestamp, [vp])]$ 
24:      end if
25:    end for
26:    return  $res, vp\_set$ 
27: end function

```

---



The GSP algorithm attempts to discover sequential patterns with the minimum support which is the probability that sequences contain the patterns. The min gap, max gap, and window size parameters determine how transactions are handled. The min gap parameter causes a sequence not to support a sequential pattern, if the transactions containing this pattern are too near in time. Meanwhile, the max gap does the same if the transaction containing this pattern are too widely separated in time. The window size determines how long a subsequent transaction is treated as the same transaction. In this research, we set the min support value as 0.1, the min gap as 0, the max gap as 300, and the window size as 10. Figure 4.6 shows the process of executing GSP algorithm in Rapidminer. [5, 6, 27, 37]



**Figure 4.6.** The interface of RapidMiner about executing GSP algorithm.

## 5 IMPLEMENTING PYTHON METHODS

This chapter introduces the second way to execute EDM, which is extracting information by Python scripts. The basic steps of implementing Python scripts are as follows:

- 1 Preparing useful Moodle log column information and generating tables that will be used as the input files of Python scripts.
- 2 Coding the Python scripts by research requirements.
- 3 Executing Python methods to obtain potential learning behaviour information.

Table 5.1 lists the six extracted Moodle log variables which are used as input data for Python methods. Among them, the information variable, which records title of resources (name of virtual patients is recorded in information variable as resource title), is mainly focused on.

**Table 5.1.** Six Moodle log variables is used as input data for Python methods.

user_id	starting_time	next_starting_time	duration_minutes	information(vp)	row_id
---------	---------------	--------------------	------------------	-----------------	--------

Section 5.1 proposes the way to extract frequency that each VP viewed by students. Section 5.2 introduces the detailed process of how to utilize Python scripts to extract virtual patient sessions in an appointed duration which is a reasonable time interval of a viewing process. In Section 5.3, the implementation idea of how to extract each VP's relationship tables is given a detailed explanation.

## 5.1 Extracting Frequencies

A summary table of the frequency that each Virtual Patient browsed by students can be generated by executing Appendix 3. The Algorithm 2 is used to generate the frequency that each VP browsed by students. There are two steps to obtain the frequency. Firstly, the Python script generates the strings which are combinations of user id and VP, as well as sets the combined strings as keys. When reading each row from input file, the number of combined string's frequency is setted as the value of the corresponding keys. Secondly, when generating a frequency dictionary, the frequency is retrieved according to the student id list and VP list given in the script file and assigned according to combined string's value that mentioned before. Through these two steps, all the results are finally putted into a result dictionary.

---

**Algorithm 2** Generate frequency that each VP browsed by students

---

**Input:** *file\_path*, the path of input file; *student\_id\_list*, the list of target student ids; *vp\_list*, the list of target VPs

**Output:** *res*, a list of each student's result dictionary which stores frequency

```

1: function get_frequency(file_path, student_id_list, vp_list)
2:   res ← empty list //Initialize the variable for output
3:   f ← open_file(file_path) //Open input file
4:   frequency_dic ← empty dictionary //Initialize the variable for storing frequencies
5:   for each row in f do //Read each row from input file
6:     user_id ← user id from row
7:     vp ← VP from row
8:     if row is the header row then //Deal with the header row
9:       continue
10:    end if
11:    key ← user_id + vp
12:    if key is already in frequency_dic then
13:      frequency_dic[key] ← frequency_dic[key]+1
14:    else
15:      frequency_dic[key] ← 1
16:    end if
17:  end for
18:  for each vp in vp_list do
19:    frequency_dic_of_one_student ← empty dictionary
20:    for each student_id in student_id_list do
21:      search_name ← student_id + vp
22:      if search_name in frequency_dic's keys then
23:        frequency_dic_of_one_student[search_name] ← ←
          frequency_dic[search_name]
24:      else
25:        frequency_dic_of_one_student[search_name] ← 0
26:      end if
27:    end for
28:    append frequency_dic_of_one_student into res]
29:  end for
30:  return res
31: end function

```

---

## 5.2 Extracting Sessions

The script given in Appendix 1 is used to gain all the Virtual Patients which are browsed by students. The script is used to extract all VPs viewed by each student and to store them in a dictionary. It reads one row from input file and checks whether the user id of this row already exists in the dictionary. If the user id exists, the VP will be appended to the end of the corresponding user id's list. If the user id does not exist, a new list that contains this VP will be mapped to the user id in the dictionary. Algorithm 3 is the pseudocode of the whole process to extract all VPs viewed by each student.

---

**Algorithm 3** Extract all VPs viewed by each student

---

**Input:** *file\_path*, the path of input file

**Output:** *res*, a dictionary of all viewed VPs and their related users

```

1: function get_vps_each_student_viewed(file_path)
2:   res ← empty dictionary //Initialize the variable for output
3:   f ← open_file(file_path) //Open input file
4:   for each row in f do //Read each row from input file
5:     user_id ← user id from row
6:     vp ← VP from row
7:     if row is the header row then //Deal with the header row
8:       continue
9:     end if
10:    if user_id is already in res then
11:      Append vp into res[user_id]
12:    else
13:      res[user_id] ← [vp]
14:    end if
15:  end for
16:  return res
17: end function

```

---

In order to analyse students' learning behaviour, reasonable duration of once online learning is assumed from 1 to 5 hours. Algorithm 4 shows the process of how to gain virtual patient sessions happening in a reasonable time interval, meanwhile, its whole Python script is recoded as shown in Appendix 2. When the Python script is executed locally, an appointed duration which means a reasonable time interval of each learning session

should be submitted, e.g., five hours. The script reads one row from input file every time and checks whether the user id of this row exists in the result dictionary's keys. If the user id exists, a new session will be generated when the time of this row exceeds the appointed duration, or modifying an existed session when the time of this row is in the range of appointed duration. If the user id does not exist, a new session will be generated. Finally, both of the results are putted into the result dictionary.

---

**Algorithm 4** Extract sessions that are in certain duration for each user

---

**Input:** *file\_path*, the path of input file; *duration*, the duration of each session in seconds

**Output:** *res*, a dictionary of all qualified sessions and their users

```

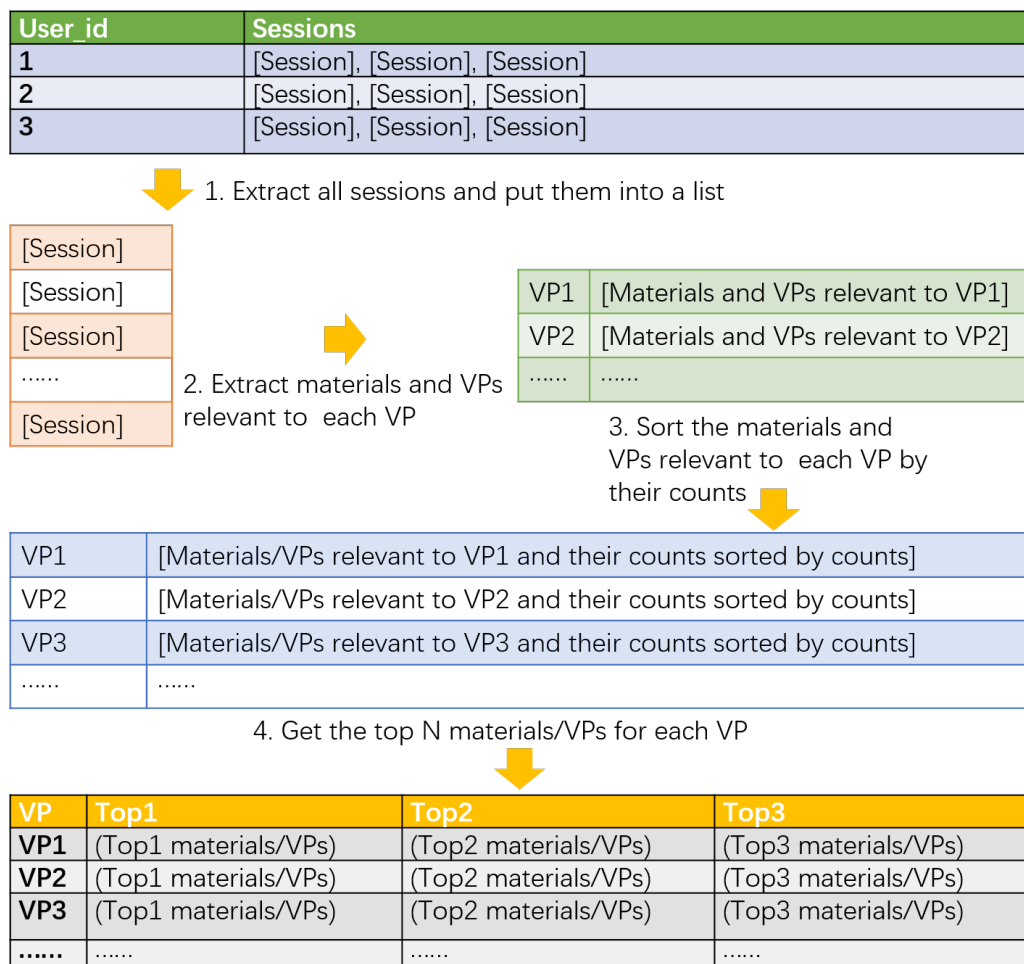
1: function get_sessions_in_certain_duration(file_path, duration)
2:   res  $\leftarrow$  empty dictionary //Initialize the variable for output
3:   f  $\leftarrow$  open_file(file_path) //Open input file
4:   for each row in f do //Read each row from input file
5:     user_id  $\leftarrow$  user id from row
6:     vp  $\leftarrow$  VP from row
7:     if row is the header row then //Deal with the header row
8:       continue
9:     end if
10:    if user_id is already in res then
11:      if the starting time of this row - the starting time of last row  $\leq$  duration
12:        then
13:          append vp into last session in res[user_id]
14:        else
15:          append [vp] into res[user_id]
16:        end if
17:      else
18:        res[user_id]  $\leftarrow$  [[vp]]
19:      end if
20:    end for
21:  return res
22: end function

```

---

### 5.3 Extracting Relationships

The implementation idea, of how to obtain each VP's top N most relative VPs or materials as well as frequency, is illustrated by Figure 5.1. There are four steps in this process: extracting all sessions and putting them into a list, extracting materials and VPs related to each VP, sorting the materials and VPs by their counts, and getting the top N most relative VPs or materials. The detailed description of each step are introduced as follows and the number of the most relative materials or VPs is assumed to be five in this section. Appendix 4 shows the full and detailed implementation of the Python script.



**Figure 5.1.** The implementation idea of extracting relationship tables.

**Step 1** – Extract all sessions and put them into a list.

This step is used to extract all sessions of a CSV file and put them into a list. The CSV file is the extracted session file from Section 5.2 and the format of the CSV file is shown in Table 5.2. One or more sessions are mapped into each `user_id`. The `user_id` variable does not be needed in this step, so only the second column called sessions is considered. This step reads CSV file and puts all sessions, except the header row, into a list. The pseudocode of this function is shown in Algorithm 5. After the function called, the example of output list is shown in Table 5.3.

**Table 5.2.** The format of the original CSV file obtained from Section 5.2.

User_id	Session
1	[VP1, Material1, Material2], [VP2, Material2, Material3], [VP1, VP3, Material1]
2	[VP2, Material1]
3	[VP1, VP2, Material1],[VP1, VP3, VP7],[Material1, Material3]
...	...

---

**Algorithm 5** Extract all sessions and put them into a list

---

**Input:** *file\_path*, the path of input file

**Output:** *res*, a list of all sessions

```

1: function read_csv(file_path)
2:   res ← empty list //Initialize the variable for output
3:   f ← open_file(file_path) //Open input file
4:   for each row in f do //Read each row from input file
5:     if row is the header row then //Deal with the header row
6:       continue
7:     end if
8:     if row[1] is not null then
9:       Append row[1] into res
10:    end if
11:  end for
12:  return res
13: end function

```

---



**Table 5.3.** Example of output in the step 1 which extracts all sessions and puts them into a list.

---

[VP1, Material1, Material2]
[VP2, Material2, Material3]
[VP1, VP3, Material1]
[VP2, Material1]
[VP1, VP2, Material1]
[VP1, VP3, VP7]
[Material1, Material3]
...

---

**Step 2 – Extract materials/VPs related to each VP.**

This step is used to generate materials or VPs related to each VP. The relationship degree, whether each VP is related to a material or VP, depends on whether they appear at the same session in Table 5.3. The pseudocode of this function is shown as Algorithm 6. For each session of input list, the function puts all VPs, which appearing at the input list, into a `vp_list`. When the input session is scanned, the items appearing at the same session will be mapped to the corresponding VPs that are in `vp_list`. Finally, the result dictionary will be generated after all sessions are scanned. In this experimental step, when the results of step 1 is inputted, the output dictionary is shown as Table 5.4.

**Table 5.4.** Example of output in the step 2 which extracts materials/VPs related to each VP.

---

Virtual Patient	Session
VP1	[VP1, Material1, Material2, VP1, VP3, Material1, VP1, VP2, Material1, VP1, VP3, VP7]
VP2	[VP2, Material2, Material3, VP2, Material1, VP1, VP2, Material1, ]
VP3	[VP1, VP3, Material1, VP1, VP3, VP7]
VP7	[VP1, VP3, VP7]
...	...

---

---

**Algorithm 6** Extract materials and VPs related to each VP
 

---

**Input:** *sessions*, the sessions extracted in step one

**Output:** *res*, a dictionary of materials/VPs related to different VPs

```

1: function process_vp(sessions)
2:   res ← empty dictionary //Initialize the variable for output
3:   for each session in sessions do
4:     vp_list ← empty list //Initialize the variable for store VPs
5:     material_and_vp_list ← empty list //Initialize the variable for store VPs/-
      materials
6:     for each part in session do
7:       if part is a VP then //If part is a VP, put it into vp_list
8:         append part into vp_list
9:       end if
10:      append part into material_and_vp_list //Copy parts in the session
11:     end for
12:     for each vp in vp_list do //Deal with result dictionary
13:       if res[vp] is existed then ///If vp is a key in res already
14:         extend res[vp] with material_and_vp_list
15:       else
16:         res[vp] ← material_and_vp_list
17:       end if
18:     end for
19:   end for
20:   return res
21: end function

```

---

**Step 3** – Sort the materials/VPs related to each VP by their counts.

This step is used to sort the materials or VPs related to each VP by their counts. Algorithm 7 is the pseudocode of Python function for this aim. Firstly, the function generates lists, of which the pair of VPs/materials and their counts is in descending order. Secondly, the function puts the materials/VPs of the same count (the  $i_{th}$  most related item) together. Finally, a dictionary of materials/VPs and their counts related to each VP will be generated as Table 5.5. Each list of VPs has several tuples, each tuple has VP/material's count and name.

---

**Algorithm 7** Sort the materials and VPs related to each VP by their counts
 

---

**Input:**  $vp\_material$ , the dictionary of all VPs and their related VPs/materials generated in step two;  $top\_num$ , the number for output quantity

**Output:**  $res$ , a dictionary of materials/VPs and their counts related to different VPs

```

1: function process_counter( $vp\_material, top\_num$ )
2:    $res \leftarrow$  empty dictionary //Initialize the variable for output
3:   for each  $vp$  in  $vp\_material$ 's keys do
4:      $top\_list \leftarrow$  empty list //Initialize the variable for store all top VPs
5:      $top\_i\_list \leftarrow$  empty list //Initialize the variable for store top i VPs
6:      $counter \leftarrow$  the VPs/materials and their counts from the most common to the
       least
7:      $i \leftarrow 0$  //The variable is used for count top N
8:     for each  $m$  in  $counter$  do
9:       if  $i \geq top\_num$  then //Already got  $top\_num$  result
10:        break
11:      end if
12:       $top\_i\_list \leftarrow$  the materials/VPs of same count(the  $i_{th}$  most common item)
13:      if  $top\_i\_list$  is not null then
14:        append  $top\_i\_list$  into  $top\_list$ 
15:      end if
16:    end for
17:     $res[vp] \leftarrow top\_list$ 
18:  end for
19:  return  $res$ 
20: end function

```

---

**Table 5.5.** Example of output in the step 3 which sorts the materials/VPs related to each VP by their counts.

Virtual Patient	Session
VP1	[[3, Material1], [2, VP3], [(1, Material2), (1, VP2), (1, VP7)]]
VP2	[[2, Material1], [(1, Material2), (1, Material3), (1, VP1)]]
VP3	[[2, VP1], [(1, Material1), (1, VP7)]]
VP7	[[1, VP1], (1, VP3)]]
...	...

---

**Step 4 – Get the top N materials/VPs for each VP.**

After step 3, the top N materials or VPs for each VP are generated. This step is used to output the result into a CSV file as Table 5.6.

**Table 5.6.** Example of final result which contains the top N materials/VPs for each VP.

<b>VP</b>	<b>Top1</b>	<b>Top2</b>	<b>Top3</b>
VP1	(3, Material1)	(2, VP3)	(1, Material2), (1, VP2), (1, VP7)
VP2	(2, Material1)	(1, Material2), (1, Material3), (1, VP1)	/
VP3	(2, VP1)	(1, Material1), (1, VP7)	/
VP7	(1, VP1), (1, VP3)	/	/
...	...	...	...

## 6 RESEARCH RESULTS

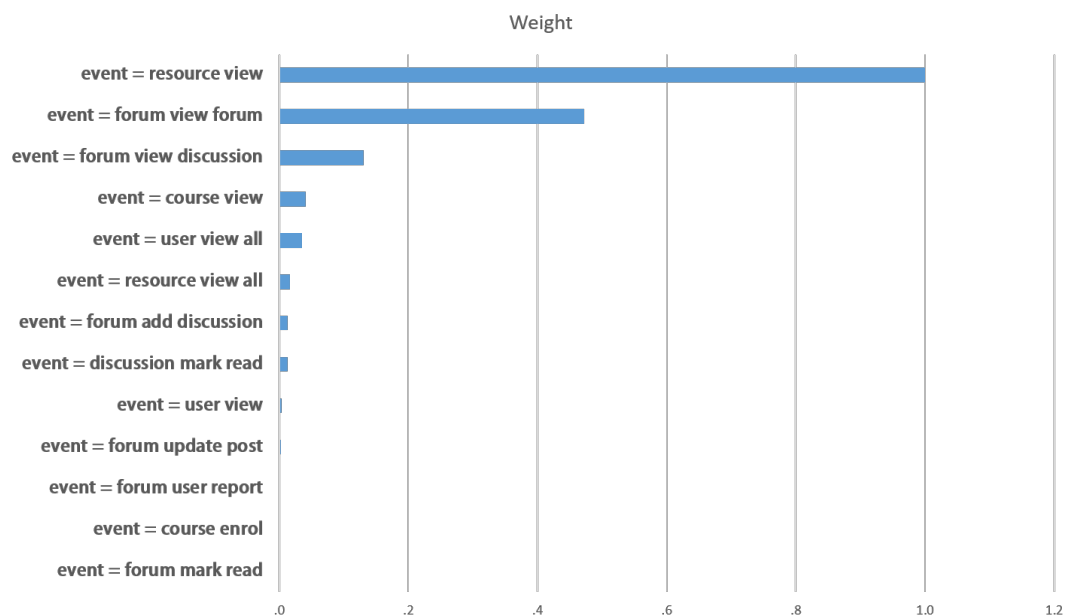
This chapter introduces the experimental results of this thesis. Section 6.1 utilizes the attribute weighting algorithms to rank all of the course events on Moodle and obtains the sort of events on which students are willing to spend more time. Section 6.2 illustrates the overall usage of Virtual Patients and the overall usage of the numbers of active students' views. Furthermore, Section 6.3 explains the result from generalized sequential patterns algorithm and uses virtual patient Terttu S as an example to analyse the result. Finally, Section 6.4 shows the results of frequencies, sessions and relationship tables by implementing Python scripts.

### 6.1 Attribute Weighting Results

In order to calculate the attribute weighting, the variable 'duration\_minutes' is setted as the 'label' role and this thesis uses '10' as the threshold to indicate whether the event is effective enough to be regarded as a valid event. The threshold classifies the value of 'duration\_minutes' into two different groups which are 'Yes' and 'No'. The values of variable 'event' are different kinds of activities happening on Moodle and all of the activities will be ranked by their corresponding weight. Table 6.1 shows the results of attribute weight by information gain and Relief algorithms. Figure 6.1 intuitively shows the result of attribute weighting by information gain by a bar graph, since the result of information gain is more informative than the other algorithm's result. Except that event of course enrol is a necessary action for each student to participate in this Moodle course, this result illustrates that resource view event has the main role among all of the Moodle activities in the medical course, while course view takes the second place in this ranking. Then, there are two events related to forum taking the third and fourth places, namely view forum and view discuss. However, other forum related events, such as add discussion, update post and user report have seldom been used by students, thus the teachers should adjust the content of forum and enhance the usage of it.

**Table 6.1.** Result of attribute weighting by information gain and Relief for which the variable 'duration\_minutes' is set as the 'label' role and the values of variable 'event' as ranked attributes.

Event	Information Gain	Relief
course enrol	0.002	1
resource view	1	0.332
courser view	0.041	0.668
forum view forum	0.472	0
forum view discussion	0.131	0
user view all	0.035	0
resource view all	0.017	0
discussion mark read	0.013	0
forum add discussion	0.013	0
user view	0.004	0
forum user report	0.002	0
forum update post	0.002	0
forum mark read	0.001	0



**Figure 6.1.** Bar graph result of attribute weighting by information gain.

## 6.2 Summary Table Results

According to Excel operation of Moodle log data, summary tables of Virtual Patients and active students are gained. In Table 6.2, the numbers of the view times for each VP in the whole course are summarized. The numbers of views in Table 6.2 means the numbers of browsing times for the all students who view resources of VPs regardless of the length of action duration. Overall, the total number of viewing resources of VPs is 721.

The range of numbers of views for each student is between 1 and 43. With the consideration of actual online viewing situation, the reasonableness of a viewing action's duration is proposed to make sure the authenticity and effectiveness of each viewing action. We consider viewing action that lasts less than 3 minutes as glance, meanwhile more than 440 minutes as invalid. When we narrow the duration of each viewing action into a time interval between 3 and 440 minutes, the numbers of effective view for each student become reduced and the new range of them is between 0 and 14. Moreover, in order to obtain the most meaningful and representative student viewing information, this thesis considers students who view VPs more than nine times at the whole course as active students, regardless of the length of viewing. Thus, in the active students' frequency summary Table 6.3, the frequency means the number of effective view ( the duration of effective view is between 3 and 440 minutes ) for each active student who views VPs more than nine times. From Table 6.3, the top three most active and representative students are Moodle users whose user\_id are 295, 110 and 73.

**Table 6.2.** The numbers of the views for each Virtual Patient.

<b>Virtual Patient</b>	<b>Number of Views</b>
VP Bertta Berlin	28
VP Laila Lumme	34
VP Pekka Oikarinen	30
VP Tuomas Tammi	28
VP Veikko Vinha	47
VP Amalia V	1
VP Carla M	23
VP George F	23
VP Heikki H	30
VP Jennifer P	27
VP Juhani M	24
VP Kauko R	46
VP Kerttu V.	37
VP Matti N	32
VP Max S	28
VP Merja P	23
VP Petteri V	32
VP Sanna A	53
VP Selma S	35
VP Shirley C	33
VP Sirkka S	28
VP Terttu S	79



**Table 6.3.** Active students frequency summary table in which the duration of each viewing action is between 3 and 440 minutes.

<b>Active Student</b>	<b>Frequency</b>
14	1
21	2
22	6
24	2
46	4
50	6
58	9
73	10
95	1
110	12
111	6
114	1
128	1
133	2
156	4
174	6
215	1
230	4
232	5
242	9
259	2
265	4
269	8
295	14
318	3
323	6

### 6.3 Generalized Sequential Patterns Results

When the minimum support is setted as 0.1, there are vase quantities of sequences generated by GSP algorithm which are difficult to manage manually. Here we analyse the most frequent VP, Terttu S, as an example and count all the other VPs or materials appearing at the same sequence with it. Table 6.4 summarizes all the teaching materials (including VPs) appearing at the same sequence with the minimum support, which means students choose to view the teaching materials listed in the table after viewing the content of VP Terttu S within 5 hours (5 hours is the value of parameter max gap of GSP algorithm). Table 6.5 summarized all the teaching materials appearing at the same transaction with VP Terttu S, which means students view the teaching materials and VP Terttu S together on Moodle within 10 minutes (10 minutes is the value of parameter window size of GSP).

**Table 6.4.** The materials or VPs appearing at the same sequences with VP Terttu S but in the different transactions, and the corresponding supports.

Teaching Material	Support
	0.258741259
	0.118881119
2011 Antimicrobials I Teacher1	0.258741259
	0.125874126
	0.223776224
Antibiotic Effect Mechanisms Teacher2 2011	0.195804196
	0.237762238
	0.125874126
Antimicrobial treatment of diarrhea Teacher2 2011	0.125874126
	0.104895105
	0.244755245
	0.125874126
Antiviral medication, Teacher3, November 8, 2011	0.104895105
	0.104895105
	0.20979021
Bacterial genetics 2011 Teacher2	0.104895105
	0.244755245
Bacterial Resistance Mechanisms 2011 Teacher2	0.118881119
	0.195804196
Dental and facial infections 2011 Teacher4	0.195804196

	0.20979021
Diagnosis of Sexual Diseases Teacher2 2011	0.160839161
	0.111888112
Exercise RT1 Working Paper Teacher2	0.13986014
	0.230769231
Hospital Kidney Infectious Bacteria 2011 Teacher2	0.118881119
	0.111888112
Infectious Disease Law Sturdy 2011	0.237762238
	0.111888112
	0.181818182
Interview and study of an adult infectious patient	0.167832168
	0.146853147
	0.167832168
Investigating an Infectionist Student's Guide 2010	0.160839161
Microbiological Diagnostics - Power Point 2010 Teacher2	0.104895105
	0.216783217
RT2 Predecessor. Note vti and sinusitis on your own sideboards!	0.160839161
	0.111888112
Sensitivity Assay Material I Teacher2	0.118881119
	0.244755245
Septic infections in children by Teacher5 2011	0.118881119
	0.118881119
	0.244755245
Sexual Diseases Teacher6 2011	0.111888112
	0.258741259
Skin infections Teacher7 2011	0.125874126
	0.104895105
	0.202797203
Toxins Teacher2 2011	0.104895105
	0.230769231
Treatment-related infections Teacher8 2010	0.111888112
Virtual patient Sanna A	0.111888112
	0.230769231
Virus Hepatitis Diagnostics 2011 Teacher2	

	0.132867133
	0.104895105
	0.251748252
Zoonoses Teacher2 2011	0.125874126
	0.118881119

**Table 6.5.** The materials or VPs appearing at the same sequences with VP Terttu S and in the same transactions, as well as the corresponding supports.

Teaching Material	Support
	0.174825175
	0.125874126
Virtual patient Kauko R	0.118881119
	0.111888112
	0.104895105
Inhibition of protein Synthesis Makrolides and tetracycline mechanism of action	0.125874126
	0.104895105

## 6.4 Python Method Results

This section introduces three types of tables acquired from the data:

**Frequency Table** – Frequency table records the frequency that each Virtual Patient was viewed by students. Table 6.6 is a matrix in which each VP is considered as the rows and student id as the columns. Given a specific VP (row<sub>*i*</sub>) and student id (column<sub>*j*</sub>), the item in the matrix denoted by  $a_{ij}$  is the frequency that the student views the VP during the whole course. Even though, almost all of the student have viewed VPs during course, the number of single VP's view times is small. Two thirds of the students did not view every VP during the whole course. Table 6.6 is part of the result of the frequency that each VP was viewed by 22 students.

**Table 6.6.** Part of the summary table of the frequency that each Virtual Patient was viewed by students.

Virtual Patients	4	5	9	14	17	21	22	24	34	46	50	58	66	71	73	77	79	81	84	90	95	97
VP Bertta Berlin	0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	1	0
VP Laila Lumme	0	0	0	0	0	0	2	2	0	1	0	1	0	0	1	0	0	0	0	0	1	0
VP Pekka Oikarinen	0	0	0	0	0	0	1	2	0	1	0	1	0	0	1	0	0	0	0	0	1	0
VP Tuomas Tammi	0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	1	0	0	0	1	0
VP Veikko VInha	0	0	0	1	0	0	1	1	0	2	0	1	0	0	1	1	0	0	0	0	1	0
VP Amalia V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
VP Carla M	0	0	0	0	0	0	0	3	0	1	1	2	0	0	1	0	0	0	0	0	0	0
VP George F	0	0	0	1	0	1	0	2	0	1	1	1	0	0	1	1	0	0	0	0	0	0
VP Heikki H	0	0	0	0	0	0	0	3	0	1	1	1	0	0	3	1	0	0	0	0	2	0
VP Jennifer P	0	0	0	0	0	1	0	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0
VP Juhani M	0	0	0	0	0	1	1	1	1	0	0	1	1	0	1	0	0	0	1	0	0	0
VP Kauko R	2	0	0	1	1	1	1	1	0	0	0	2	0	1	1	1	0	1	0	0	0	0
VP Kerttu V.	1	0	0	0	0	1	2	2	0	0	1	1	0	0	1	0	0	2	0	1	0	0
VP Matti N	0	0	0	1	0	1	2	2	0	0	0	1	0	0	1	0	0	0	0	0	0	0
VP Max S	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	2	0	0	0
VP Merja P	0	0	0	1	0	1	0	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0
VP Petteri V	0	0	0	1	0	1	1	2	0	0	0	1	0	0	2	0	0	0	1	0	1	0
VP Sanna A	0	1	0	1	0	1	2	2	0	2	1	2	0	0	1	0	0	0	0	0	0	1
VP Selma S	0	0	1	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
VP Shirley C	0	0	0	0	0	0	1	3	0	1	1	1	0	0	1	0	0	0	0	0	0	0
VP Sirkka S	0	0	0	0	0	1	0	2	0	0	1	1	0	0	1	0	0	0	0	0	1	0
VP Terttu S	0	0	0	1	1	1	2	9	1	1	0	5	1	1	1	1	0	1	0	1	0	2

**Session Table** – Virtual patient sessions are the foundation utilized to finally extract relationship tables. The relationship tables illustrate the correlation of that VPs appear at the same duration which is the time interval of 5 hours. Thus, VP sessions collecting the VPs appearing at a certain duration are the foundation for extracting relationship tables. Table 6.7 lists part of results that all the VPs viewed by each student while the complete results are recorded in Appendix 5. It is clear to know the usage of VP for each student and evaluate the learning behaviour based on the results. However, because only one student failed in this course and the other students were all evaluated with 'passed' for final score, it is impossible to analyse the relationship between the usage of VP and students' final course grade. Moreover, Table 6.8 shows part of the VP sessions taking place in 5 hours by each student. Similarly, the complete table will be present in Appendix 6.

**Table 6.7.** Ordered Virtual Patients which are browsed by each user.

User_id	Virtual Patients
4	VP Kauko R, VP Kauko R, VP Kerttu V.
5	VP Max S, VP Sanna A
9	VP Selma S
14	VP Terttu S, VP Kauko R, VP Petteri V, VP Matti N, VP Sanna A, VP George F, VP Merja P, VP Max S, VP Veikko VInha
17	VP Terttu S, VP Kauko R
21	VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Merja P, VP George F, VP Sanna A, VP Max S VP Juhani M, VP Jennifer P, VP Sirkka S
22	VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Terttu S, VP Terttu S, VP Kauko R, VP Kerttu V. VP Kerttu V., VP Matti N, VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Shirley C, VP Sanna A, VP Laila Lumme
...	...

**Table 6.8.** Virtual patient sessions taking place in 5 hours.

User_id	Virtual Patient Sessions
4	[VP Kauko R], [VP Kauko R, VP Kerttu V.]
5	[VP Max S], [VP Sanna A]
9	[VP Selma S]
14	[VP Terttu S, VP Kauko R], [VP Petteri V, VP Matti N], [VP Sanna A, VP George F, VP Merja P], [VP Max S], [VP Veikko VInha]
17	[VP Terttu S, VP Kauko R]
21	[VP Terttu S, VP Kauko R], [VP Kerttu V.], [VP Matti N, VP Selma S], [VP Petteri V] [VP Merja P, VP George F, VP Sanna A, VP Max S, VP Juhani M], [VP Jennifer P, VP Sirkka S]
...	...

**Relationship Table** – Relationship table illustrates the top five most relative VPs of each VP. The relationship degree depends on whether these two VPs (or other materials) appear inside the same time interval of 5 hours for a certain student. Table 6.9 is the relationship table of all VPs. VP Terttu S is the most significant one which are frequent top 1 relative VP for the others, while VP Sanna A take the second place. Table 6.10 does the same with not only the VPs, but also the other teaching materials. It is a good analysis object to evaluate students' abilities about choosing right medical materials based on a certain VP's symptom and description.

**Table 6.9.** Top 5 relative Virtual Patients' names and frequencies for all Virtual Patients.

Virtual Patients	Top1	Top2	Top3	Top4	Top5
VP Veikko VInha	28, VP Laila Lumme	26, VP Pekka Oikarinen	25, VP Bertta Berlin	24, VP Tuomas Tammi	22, VP Terttu S
VP Heikki H	32, VP Terttu S	21, VP Carla M	20, VP Sanna A, 20, VP Sirkka S	19, VP Shirley C	18, VP George F, 18, VP Merja P
VP Tuomas Tammi	25, VP Laila Lumme	24, VP Bertta Berlin, 24, VP Pekka Oikarinen, 24, VP Veikko VInha	15, VP Terttu S	9, VP Sanna A	8, VP Heikki H
VP Kauko R	41, VP Terttu S	20, VP Kerttu V.	14, VP Sanna A	13, VP Matti N, 13, VP Petteri V	12, VP Selma S
VP Sanna A	34, VP Terttu S	28, VP Shirley C	24, VP Carla M	20, VP George F, 20, VP Heikki H	19, VP Merja P
VP Kerttu V.	36, VP Terttu S	21, VP Matti N	20, VP Kauko R	14, VP Selma S	13, VP Petteri V, 13, VP Sanna A
VP George F	23, VP Terttu S	20, VP Sanna A	18, VP Heikki H	16, VP Carla M	15, VP Shirley C
VP Juhani M	20, VP Terttu S	15, VP Sanna A	14, VP Max S	13, VP Petteri V	12, VP Matti N
VP Terttu S	41, VP Kauko R	38, VP Shirley C	36, VP Kerttu V.	35, VP Matti N	34, VP Carla M, 34, VP Sanna A
VP Matti N	35, VP Terttu S	22, VP Selma S	21, VP Kerttu V.	16, VP Petteri V	15, VP Sanna A
VP Petteri V	30, VP Terttu S	18, VP Sanna A	16, VP Matti N, 16, VP Shirley C	15, VP Carla M, 15, VP Heikki H, 15, VP Selma S	13, VP Juhani M, 13, VP Kauko R, 13, VP Kerttu V.
VP Laila Lumme	28, VP Pekka Oikarinen, 28, VP Veikko VInha	26, VP Bertta Berlin	25, VP Tuomas Tammi, 25, VP Terttu S	14, VP Sanna A	12, VP Heikki H, 12, VP Matti N
VP Amalia V	-1, 'N/A	-1, 'N/A	-1, 'N/A	-1, 'N/A	-1, 'N/A
VP Selma S	22, VP Matti N, 22, VP Terttu S	15, VP Petteri V	14, VP Kerttu V.	12, VP Kauko R	9, VP Juhani M, 9, VP Sanna A
VP Max S	18, VP Terttu S	17, VP Sanna A	14, VP Heikki H, 14, VP Juhani M	11, VP Carla M, 11, VP Matti N, 11, VP Petteri V	10, VP Shirley C
VP Pekka Oikari- nen	28, VP Laila Lumme	26, VP Bertta Berlin, 26, VP Veikko VInha	25, VP Terttu S	24, VP Tuomas Tammi	14, VP Heikki H
VP Merja P	19, VP Sanna A	18, VP Heikki H	16, VP Terttu S	15, VP Carla M, 15, VP Shirley C	12, VP George F
VP Bertta Berlin	26, VP Laila Lumme, 26, VP Pekka Oikarinen	25, VP Veikko VInha	24, VP Tuomas Tammi	15, VP Terttu S	10, VP Sanna A
VP Sirkka S	21, VP Terttu S	20, VP Heikki H	15, VP Sanna A	14, VP Carla M	13, VP George F, 13, VP Shirley C
VP Shirley C	38, VP Terttu S	28, VP Sanna A	24, VP Carla M	19, VP Heikki H	16, VP Petteri V
VP Carla M	34, VP Terttu S	24, VP Sanna A, 24, VP Shirley C	21, VP Heikki H	16, VP George F	15, VP Merja P, 15, VP Petteri V



**Table 6.10.** Top 5 relative materials (including Virtual Patients) and frequencies for all Virtual Patients.

Virtual Patient	Top1	Top2	Top3	Top4	Top5
VP Veikko VInha	30, Upper respiratory tract infections	28, VP Laila Lumme	26, VP Pekka Oikarinen	25, VP Bertta Berlin	24, VP Tuomas Tammi
VP George F	23, VP Terttu S	20, VP Sanna A	18, VP Heikki H	16, VP Carla M	15, VP Shirley C
VP Pekka Oikari- nen	29, Upper respiratory tract infections	28, VP Laila Lumme	26, VP Bertta Berlin, 26, VP Veikko VInha	25, VP Terttu S	24, VP Tuomas Tammi
VP Kauko R	41, VP Terttu S	28, Inhibition of protein Synthesis Makrolides and tetracycline mechanism of action	20, VP Kerttu V.	17, Antibiotic Effect Mechanisms Teacher2 2011	16, Diagnosis of Sexual Diseases Teacher2 2011
VP Sanna A	34, VP Terttu S	28, VP Shirley C	24, VP Carla M	20, VP George F, 20, VP Heikki H	19, 2011 Antimicrobials I Teacher1, 19, VP Merja P
VP Kerttu V.	36, VP Terttu S	21, VP Matti N	20, VP Kauko R	18, Septic infections in children by Teacher5 2011	16, antibiotic animation
VP Heikki H	32, VP Terttu S	21, VP Carla M	20, VP Sanna A, 20, VP Sirkka S	19, VP Shirley C	18, VP George F, 18, VP Merja P
VP Max S	18, VP Terttu S	17, VP Sanna A	14, VP Heikki H, 14, VP Juhani M	11, VP Carla M, 11, VP Matti N, 11, VP Petteri V	10, 2011 Antimicrobials I Teacher1, 10, RT2 Predecessor. Note vti and sinusitis on your own sideboards!, 10, VP Shirley C
VP Terttu S	41, VP Kauko R	38, VP Shirley C	36, VP Kerttu V.	35, VP Matti N	34, VP Carla M, 34, VP Sanna A
VP Matti N	35, VP Terttu S	22, VP Selma S	21, VP Kerttu V.	16, VP Petteri V	15, VP Sanna A
VP Petteri V	30, VP Terttu S	19, Zoonoses Teacher2 2011	18, VP Sanna A	16, VP Matti N, 16, VP Shirley C	15, VP Carla M, 15, VP Heikki H, 15, VP Selma S
VP Laila Lumme	30, Upper respiratory tract infections	28, VP Pekka Oikarinen, 28, VP Veikko VInha	26, VP Bertta Berlin	25, VP Tuomas Tammi, 25, VP Terttu S	17, Microbiological Diagnostics - Power Point 2010 Teacher2
VP Amalia V	-1, N/A	-1, N/A	-1, N/A	-1, N/A	-1, N/A
VP Selma S	22, VP Matti N, 22, VP Terttu S	15, VP Petteri V	14, VP Kerttu V.	13, Inhibition of nucleic acid synthesis	12, VP Kauko R
VP Merja P	19, VP Sanna A	18, VP Heikki H	16, VP Terttu S	15, VP Carla M, 15, VP Shirley C	12, VP George F
VP Tuomas Tammi	25, VP Laila Lumme	24, VP Bertta Berlin, 24, VP Pekka Oikarinen, 24, VP Veikko VInha, 24, Upper respiratory tract infections	15, VP Terttu S	13, Microbiological Diagnostics - Power Point 2010 Teacher2	11, 2011 Antimicrobials II Teacher1, 11, Hospital Kidney Infectious Bacteria 2011 Teacher2

VP Bertta Berlin	26, VP Laila Lumme, 26, VP Pekka Oikarinen	25, VP Veikko Vinha, 25, Upper respiratory tract infections	24, VP Tuomas Tammi	17, Microbiological Diagnostics - Power Point 2010 Teacher2	15, Hospital Kidney Infectious Bacteria 2011 Teacher2, 15, VP Terttu S
VP Sirkka S	21, VP Terttu S	20, VP Heikki H	15, VP Sanna A	14, VP Carla M	13, VP George F, 13, VP Shirley C
VP Juhani M	20, VP Terttu S	15, VP Sanna A	14, VP Max S	13, VP Petteri V	12, VP Matti N
VP Shirley C	38, VP Terttu S	28, VP Sanna A	24, VP Carla M	19, VP Heikki H	16, VP Petteri V
VP Carla M	34, VP Terttu S	24, VP Sanna A, 24, VP Shirley C	21, VP Heikki H	16, VP George F	15, VP Merja P, 15, VP Petteri V

---

## 7 CONCLUSION

This thesis conducts EDM on a medical course to analyse learning behaviour about PBL method, especially on the usage of VPs which contain patients' condition descriptions and are used as online teaching materials on Moodle. The usage of VPs reveals how students follow PBL method on the medical course. It is a significant topic because the usage of VP help students to obtain practical skills. In order to analyse Moodle learning behaviour about VPs, two kinds of research ways inspired from EDM methods are applied in this thesis. One is applying relationship mining and prediction approach to use attribute weighting and GSP algorithms in Rapidminer in order to obtain learning patterns. The other is implementing Python scripts to obtain patterns for human judgment.

The aims of this thesis are analysing learning behaviour, providing teachers with suggestions, and implementing Python scripts to handle the Moodle logs. Firstly, in this course case, VP Terttu S is confirmed as the most popular one who becomes as the top 1 relative VP for many others. Relationship table illustrates the relationships among VPs and other teaching materials, which is a good analysis object to evaluate students' abilities about choosing right medical contents based on VPs' symptoms and descriptions. Secondly, although the percentage of students who have been viewed VPs is pretty good, the coverage of students who view all single VP is not ideal. It illustrates that VP has been accepted by the most of students and the intention to follow PBL method to arrange this course has a good start. However, it is necessary for teachers to promote the usage of VP by enhancing the contents or issuing instructions. The other situation founded is that forum acts more like a reading material for students than other important characteristic called group work in PBL. Students do lots of view actions on forum but seldom do discussion, communication, submission and cooperation actions. Thus, the instruction on how to use forum and improvement of forum form should be considered. Thirdly, the Python algorithms of how to extract frequency tables, sessions, and relationship tables have been implemented and introduced in Chapter 5 for further utilization.

There are some limitations on this research. Because Moodle just records the start time of every event, the duration values used in this research are calculated as the interval of two events instead of an accurate duration of a certain event. It will cause some errors in the final results. This research uses a time interval of 5 hours as the duration of a session, because all of the learning actions within 5 hours can be regarded as a whole reasonable learning process. Moreover, further analysis about meaning of learning behaviour on VP cannot be produced, since the knowledge from this application area requires experience of the medical education and related medical background. This thesis obtains learning

behaviour records and patterns about each VP, but the reasons, why they are viewed in this order and relationships between each VP, need more supporting information and data to be explained. Lastly, due to the performance of students in this course was evaluated with 'passed/failed' and only one student failed, the research about relationship between the usage of VP and students' final evaluation is impossible to be conducted.

The session table generated from this thesis can be recommended as a good tool for other researchers to obtain summary information of utilization of teaching materials. Because there are a large amount of results, visualisation methods should be developed to help scrutinize the results. Furthermore, all Python algorithms provided in Chapter 5 implement new tools of handling the Moodle log data to extract frequencies, sessions, and relationship tables for other researchers.

## REFERENCES

- [1] Adamo, J.-M. (2012). *Data Mining for Association Rules and Sequential Patterns: Sequential and Parallel Algorithms*. Springer Science & Business Media, Secaucus, NJ, USA.
- [2] Adams, M. and Jordan, M. (2012). Taking a problem-based learning approach to teaching ecopsychology. *Ecopsychology*, 4(2):87–93.
- [3] Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer Publishing Company, Incorporated.
- [4] Aggarwal, C. C. and Han, J. (2014). *Frequent Pattern Mining*. Springer International Publishing.
- [5] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering, 1995.*, pages 3–14. IEEE.
- [6] Akthar, F. and Hahne, C. (2012). Rapidminer 5 operator reference. *Rapid-I GmbH*.
- [7] Albanese, M. A. (2010). Problem-based learning. In *An Introduction to Medical Teaching*, pages 41–53. Springer.
- [8] Baker, R. et al. (2010). Data mining for education. *International Encyclopedia of Education*, 7(3):112–118.
- [9] Balian, R. (2004). Entropy, a protean concept. In *Poincaré Seminar 2003*, pages 119–144. Springer.
- [10] Barrows, H. S. (1986). A taxonomy of problem-based learning methods. *Medical Education*, 20(6):481–486.
- [11] Baruque, C. B., Amaral, M. A., Barcellos, A., da Silva Freitas, J. C., and Longo, C. J. (2007). Analysing users’ access logs in moodle to improve e learning. In *Proceedings of the 2007 Euro American Conference on Telematics and Information Systems*, page 72. ACM.
- [12] Casey, K., Gibson, P., and Paris, I.-S. (2010). Mining moodle to understand student behaviour. In *International Conference on Engaging Pedagogy 2010 (ICEP10)*, National University of Ireland Maynooth.
- [13] Castro, F., Vellido, A., Nebot, À., and Mugica, F. (2007). Applying data mining techniques to e-learning problems. In *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, pages 183–221. Springer.

- [14] Dolmans, D. H., De Grave, W., Wolfhagen, I. H., and Van Der Vleuten, C. P. (2005). Problem-based learning: Future challenges for educational practice and research. *Medical Education*, 39(7):732–741.
- [15] Dong, G. and Pei, J. (2007). *Sequence Data Mining*, volume 33. Springer Science & Business Media.
- [16] Han, J., Pei, J., and Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.
- [17] Hanna, M. (2004). Data mining in the e-learning domain. *Campus-Wide Information Systems*, 21(1):29–34.
- [18] Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16(3):235–266.
- [19] Hunt, E. B., Marin, J., and Stone, P. J. (1966). Experiments in induction.
- [20] Ignatov, D. I., Mitrofanova, E., Muratova, A., and Gizdatullin, D. (2015). Pattern mining and machine learning for demographic sequences. In *International Conference on Knowledge Engineering and the Semantic Web*, pages 225–239. Springer.
- [21] IVIMEDS-Partnership (Accessed April 4, 2017). *IVIMEDS*. <http://ivimeds.org.uk/>.
- [22] Jacob, J., Jha, K., Kotak, P., and Puthran, S. (2015). Educational data mining techniques and their applications. In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pages 1344–1348. IEEE.
- [23] Jovic, A., Brkic, K., and Bogunovic, N. (2014). An overview of free software tools for general data mining. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1112–1117. IEEE.
- [24] Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, pages 249–256.
- [25] Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *European Conference on Machine Learning*, pages 171–182. Springer.
- [26] Macallan, D. C., Kent, A., Holmes, S. C., Farmer, E. A., and McCrorie, P. (2009). A model of clinical problem-based learning for clinical attachments in medicine. *Medical Education*, 43(8):799–807.

- [27] Masegla, F., Poncelet, P., and Teisseire, M. (2004). Pre-processing time constraints for efficiently mining generalized sequential patterns. In *Proceedings. 11th International Symposium on Temporal Representation and Reasoning, 2004. TIME 2004.*, pages 87–95. IEEE.
- [28] Mazza, R. and Dimitrova, V. (2007). Coursevis: A graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human-Computer Studies*, 65(2):125–139.
- [29] Myszkowski, P. B., Kwasnicka, H., and Markowska-Kaczmar, U. (2008). Data mining techniques in e-learning celgrid system. In *Computer Information Systems and Industrial Management Applications, 2008. CISIM'08.*, pages 315–319. IEEE.
- [30] Nguyen, L. and Do, P. (2009). Learning concept recommendation based on sequential pattern mining. In *3rd IEEE International Conference on Digital Ecosystems and Technologies, 2009. DEST'09.*, pages 66–71. IEEE.
- [31] Poikela, E. and Nummenmaa, A. R. (2006). *Understanding Problem-based Learning*. University of Tampere.
- [32] Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., and Ventura, S. (2013). Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146.
- [33] Romero, C. and Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618.
- [34] Romero, C., Ventura, S., and García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384.
- [35] Schaefer, V. and Gonzales, E. (2013). Using problem-based learning to teach concepts for ecological restoration. *Ecological Restoration*, 31(4):412–418.
- [36] Srikant, R. and Agrawal, R. (1995). Mining generalized association rules. *Future Generation Computer Systems*, 13(2):161 – 180.
- [37] Srikant, R. and Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Advances in Database Technology-EDBT 96*, pages 1–17.
- [38] Sukhija, K., Jindal, M., and Aggarwal, N. (2016). Educational data mining towards knowledge engineering: a review state. *International Journal of Management in Education*, 10(1):65–76.

- [39] Taniar, D., Leung, C. H., Rahayu, W., and Goel, S. (2008). Parallel clustering and classification. *High-Performance Parallel Database Processing and Grid Databases*, pages 464–499.
- [40] Walldén, S. and Mäkinen, E. (2014). Educational data mining and problem-based learning. *Informatics in Education*, 13(1):141.
- [41] Wood, D. F. (2003). Abc of learning and teaching in medicine problem based learning. *BMJ*, 326:8.
- [42] Wu, A. K. and Leung, C. (2002). Evaluating learning behavior of web-based training (wbt) using web log. In *Proceedings. International Conference on Computers in Education, 2002.*, pages 736–737. IEEE.
- [43] Zaiane, O. (2001). Web usage mining for a better web-based learning environment. *Proceedings of Conference on Advanced Technology for Education, Banff, Alberta.*
- [44] Zorrilla, M. E., Menasalvas, E., Marin, D., Mora, E., and Segovia, J. (2005). Web usage mining project for improving web-based learning sites. In *International Conference on Computer Aided Systems Theory*, pages 205–210. Springer.



## Appendix 1.

### Script for Table 6.7

```
import xlrd
vp = xlrd.open_workbook ("vp_student.xlsx")
sh = vp.sheet_by_index (0)

#column 0 = id; column 4 = vp
def fun():
    dic={}
    for rx in range (1,sh.nrows):
        if sh.cell_value(rx,0) in dic.keys():
            dic[sh.cell_value(rx,0)].append(sh.cell_value(rx,4))
        else:
            dic[sh.cell_value(rx,0)] = [sh.cell_value(rx,4)]
    return dic

import csv
with open ("result111.csv", 'w') as csvfile:
    dic = fun()
    fieldnames = ['id','vps']
    writer = csv.DictWriter (csvfile, fieldnames = fieldnames)
    writer.writeheader()
    for i in dic.keys():
        writer.writerow({'id':i,'vps':dic[i]})
        # print 'id: '+str(i)
        # print 'vps: '+str(dic[i])
```

(continues)

## Appendix 2. (continued)

Script for appointed duration to gain virtual patients sessions of each student

```
import xlrd
import datetime
vp = xlrd.open_workbook ("vp_student.xlsx")
sh = vp.sheet_by_index (0)

# read input

#column 0 = id; column 4 = vp
def fun():
    dic={}
    i = 0
    itime = int(input("Enter the duration of one session in second(for
                        example: 3600 = 1h)"))

    for rx in range (1,sh.nrows):
        if sh.cell_value(rx,0) in dic.keys():
            a = datetime.datetime.strptime(sh.cell_value(rx,1), "%d.%m.
                                           %Y %H:%M")
            b = datetime.datetime.strptime(sh.cell_value(rx-1,1), "%d.%
                                           m.%Y %H:%M")

            if a-b <= datetime.timedelta(0, itime):
                dic[sh.cell_value(rx,0)][i].append (sh.cell_value(rx,4)
                                                    )

            else:
                dic[sh.cell_value(rx,0)].append([sh.cell_value(rx,4)])
                i += 1

        else:
            i =0
            dic[sh.cell_value(rx,0)] = [[sh.cell_value(rx,4)]]

    return dic

import csv
with open ("result_inputtime.csv", 'w') as csvfile:
    dic = fun()
    fieldnames = ['id','vps']
    writer = csv.DictWriter (csvfile, fieldnames = fieldnames)
    writer.writeheader()
    for i in dic.keys():
        writer.writerow({'id':i,'vps':dic[i]})
```

(continues)

## Appendix 3. (continued)

### Script for Table 6.6

```
import xlrd
import datetime
vp = xlrd.open_workbook ("vp_student.xlsx")
sh = vp.sheet_by_index (0)

# read input

#column 0 = id; column 4 = vp
def fun():
    dic={}
    i = 0
    for rx in range (1,sh.nrows):
        if str(sh.cell_value(rx,0))+':' +sh.cell_value(rx,4) in dic.keys
            ():

            dic[str(sh.cell_value(rx,0))+':' +sh.cell_value(rx,4)] +=1

        else:
            i=0
            dic[str(sh.cell_value(rx,0))+':' +sh.cell_value(rx,4)] = i+1
    return dic

import csv
with open ("result_summaryVP(vpAsRow).csv", 'w') as csvfile:
    dic = fun()
    rows = []
    listid = [4.0,5.0,9.0,14.0,17.0,21.0,22.0,24.0,34.0,46.0,50.0,58.0,
66.0,71.0,73.0,77.0,79.0,81.0,
84.0,90.0,95.0,97.0,106.0,107.0
,108.0,109.0,110.0,111.0,112.0,
113.0,114.0,115.0,128.0,130.0,
133.0,134.0,135.0,139.0,152.0,
156.0,162.0,163.0,165.0,166.0,
168.0,172.0,174.0,184.0,194.0,
204.0,206.0,208.0,211.0,215.0,
216.0,220.0,221.0,225.0,226.0,
230.0,232.0,241.0,242.0,246.0,
259.0,261.0,263.0,265.0,269.0,
272.0,275.0,281.0,283.0,288.0,
293.0,295.0,298.0,315.0,318.0,
321.0,322.0,323.0,324.0,325.0,
328.0]

    listvp = ['Virtuaalipotilas Bertta Berlin', 'Virtuaalipotilas Laila
Lumme', 'Virtuaalipotilas Pekka
Oikarinen', 'Virtuaalipotilas
```

(continues)

### Appendix 3. (continued)

```
Tuomas Tammi', 'Virtuaalipotilas
Veikko VInha', 'Virtual patient
Amalia V', 'Virtual patient
Carla M', 'Virtual patient
George F', 'Virtual patient
Heikki H', 'Virtual Patient
Jennifer P', 'Virtual patient
Juhani M', 'Virtual patient
Kauko R', 'Virtual patient
Kerttu V.', 'Virtual patient
Matti N', 'Virtual patient Max S
', 'Virtual patient Merja P', '
Virtual patient Petteri V', '
Virtual patient Sanna A', '
Virtual patient Selma S', '
Virtual patient Shirley C', '
Virtual patient Sirkka S', '
Virtual patient Terttu S']

for vp in listvp:
    row = {'vp': vp}
    for idrx in listid:
        if str(idrx) + ':' + vp in dic.keys():
            row[idrx] = dic[str(idrx) + ':' + vp]
        else:
            row[idrx] = 0
    rows.append(row)

fieldnames = ['vp'] + listid
writer = csv.DictWriter (csvfile, fieldnames = fieldnames)
writer.writeheader()
writer.writerows (rows)
```

(continues)

## Appendix 4. (continued)

### Script for Table 6.9

```
# encoding=utf8
import sys

reload(sys)
sys.setdefaultencoding('utf8')

import csv
import collections
import copy

def read_csv(file_path):
    res = []
    with open(file_path, 'rb') as f:
        reader = csv.reader(f)
        header_flag = True
        for row in reader:
            if header_flag:
                header_flag = False
                continue

            if row[1] != '[]':
                res.append(eval(row[1]))
    return res

def process_vp(objs):
    res = {}
    for obj in objs:
        for item in obj:
            vp_list = []
            material_and_vp_list = []
            for part in item:
                if part.find('Virtuaalipotilas') != -1:
                    vp_list.append(part)
                elif part.find('Virtual patient') != -1:
                    vp_list.append(part)
                material_and_vp_list.append(part)

            for vp in vp_list:
                tmp = copy.deepcopy(material_and_vp_list)
                if vp in res.keys():
                    res[vp].extend(tmp)
                else:
                    res[vp] = tmp
```

(continues)

## Appendix 4. (continued)

```
# print res
return res

def process_counter(vp_material, top_num):
    res = {}

    for vp in vp_material.keys():
        print vp,
        counter = collections.Counter(vp_material[vp]).most_common()
        counter.sort(key=lambda x:x[0])
        counter.sort(key=lambda x:x[1], reverse=True)
        i = 0
        last_count = -1
        top_list = []

        # print counter
        top_i = []
        for m in counter:
            if i >= top_num and last_count != m[1]:
                break
            if m[0] != vp:

                if last_count != m[1] and len(top_i) != 0:
                    top_list.append(top_i)
                    i += 1

                top_i = [m]
            else:
                top_i.append(m)
            last_count = m[1]
        if len(top_i) != 0:
            top_list.append(top_i)
        print top_i,
        print '======'
        res[vp] = top_list
    return res

def output(output_path, counted_dict, top_num, mode=1):
    with open(output_path, 'wb') as f:
        fieldnames = ['VP', 'Top1', 'Top2', 'Top3', 'Top4', 'Top5']
        writer = csv.DictWriter(f, fieldnames=fieldnames)
        writer.writeheader()

        for vp in counted_dict.keys():
            ref = counted_dict[vp]
            ref.extend(map(lambda x: ['N/A', -1], range(top_num+1)))
```

(continues)

## Appendix 4. (continued)

```
    if mode == 1:
        writer.writerow({'VP': vp, 'Top1':map(lambda x:x[::-1],
                                             ref[0]), 'Top2':
                                             map(lambda x:x[::-1]
                                             ], ref[1]), 'Top3':
                                             map(lambda x:x[::-1]
                                             ], ref[2]), 'Top4':
                                             map(lambda x:x[::-1]
                                             ], ref[3]), 'Top5':
                                             map(lambda x:x[::-1]
                                             ], ref[4])})

    else:
        writer.writerow({'VP': vp, 'Top1':ref[0][0], 'Top2':ref
                        [1][0], 'Top3':ref[
                        2][0], 'Top4':ref[3
                        ][0], 'Top5':ref[4]
                        [0]})

def main():
    input_file_path = 'result_inputtime(resource material).csv'
    output_file_path = 'test_output5.csv'
    top_num = 5

    objs = read_csv(input_file_path)
    vp_material = process_vp(objs)
    counted_dict = process_counter(vp_material, top_num)
    # print counted_dict
    output(output_file_path, counted_dict, top_num)

main()
```

(continues)

## Appendix 5. (continued)

**Table A5.1.** Complete table of ordered Virtual Patients which are browsed by each user.

User_id	Virtual Patient
4	VP Kauko R, VP Kauko R, VP Kerttu V.
5	VP Max S, VP Sanna A
9	VP Selma S
14	VP Terttu S, VP Kauko R, VP Petteri V, VP Matti N, VP Sanna A, VP George F, VP Merja P, VP Max S, VP Veikko VInha
17	VP Terttu S, VP Kauko R
21	VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Merja P, VP George F, VP Sanna A, VP Max S, VP Juhani M, VP Jennifer P, VP Sirkka S
22	VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Terttu S, VP Terttu S, VP Kauko R, VP Kerttu V., VP Kerttu V., VP Matti N, VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Shirley C, VP Sanna A, VP Laila Lumme
24	VP Petteri V, VP Terttu S, VP Terttu S, VP Laila Lumme, VP Terttu S, VP Veikko VInha, VP Terttu S, VP Laila Lumme, VP Terttu S, VP Tuomas Tammi, VP Terttu S, VP Bertta Berlin, VP Terttu S, VP Pekka Oikarinen, VP Pekka Oikarinen, VP Terttu S, VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sirkka S, VP Sanna A, VP Shirley C, VP Carla M, VP Shirley C, VP Carla M, VP George F, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Heikki H, VP Sirkka S, VP Heikki H, VP Heikki H, VP Jennifer P
34	VP Terttu S, VP Max S, VP Juhani M
46	VP George F, VP Sanna A, VP Sanna A, VP Shirley C, VP Carla M, VP Jennifer P, VP Merja P, VP Heikki H, VP Veikko VInha, VP Laila Lumme, VP Veikko VInha, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Terttu S
50	VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Selma S, VP Kerttu V., VP Merja P, VP Heikki H, VP Sirkka S, VP Jennifer P

(continues)



## Appendix 5. (continued)

- 58 VP Terttu S, VP Kauko R, VP Terttu S, VP Kauko R, VP Kerttu V., VP Terttu S, VP Matti N, VP Terttu S, VP Sanna A, VP Veikko VInha, VP Terttu S, VP Shirley C, VP Sanna A, VP Carla M, VP George F, VP Carla M, VP Merja P, VP Heikki H, VP Sirkka S, VP Jennifer P, VP Petteri V, VP Juhani M, VP Max S, VP Pekka Oikarinen, VP Bertta Berlin, VP Tuomas Tammi, VP Laila Lumme
- 66 VP Max S, VP Juhani M, VP Terttu S
- 71 VP Terttu S, VP Kauko R
- 73 VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Heikki H, VP Heikki H, VP Heikki H, VP Sirkka S, VP Jennifer P
- 77 VP Kauko R, VP Heikki H, VP Veikko VInha, VP Terttu S, VP George F
- 79 VP Tuomas Tammi
- 81 VP Kerttu V., VP Kerttu V., VP Terttu S, VP Kauko R
- 84 VP Petteri V, VP Juhani M, VP Max S, VP Max S
- 90 VP Terttu S, VP Kerttu V.
- 95 VP Heikki H, VP Sirkka S, VP Heikki H, VP Petteri V, VP Veikko VInha, VP Bertta Berlin, VP Laila Lumme, VP Tuomas Tammi, VP Pekka Oikarinen
- 97 VP Selma S, VP Terttu S, VP Sanna A, VP Terttu S
- 106 VP Max S, VP Sirkka S, VP Max S
- 107 VP Matti N, VP Selma S, VP Terttu S, VP Terttu S
- 108 VP Kerttu V., VP Kerttu V., VP Kauko R
- 109 VP Matti N, VP Selma S
- 110 VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Heikki H, VP Sirkka S, VP Jennifer P
- 111 VP Terttu S, VP Sanna A, VP Shirley C, VP Merja P, VP Veikko VInha, VP Petteri V, VP Terttu S, VP Tuomas Tammi, VP Bertta Berlin, VP Veikko VInha, VP Laila Lumme, VP Pekka Oikarinen

(continues)

## Appendix 5. (continued)

- 112 VP Juhani M, VP Juhani M
- 113 VP Kauko R, VP Sanna A, VP Merja P, VP Shirley C
- 114 VP Kauko R, VP Terttu S, VP Kerttu V., VP Tuomas Tammi, VP Laila Lumme, VP Veikko VInha, VP Bertta Berlin, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Shirley C, VP Merja P, VP Jennifer P, VP Sirkka S, VP Heikki H, VP Shirley C, VP Carla M
- 115 VP Tuomas Tammi
- 128 VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Shirley C, VP Carla M, VP Merja P, VP Max S, VP Carla M, VP Veikko VInha, VP Terttu S, VP Heikki H, VP Veikko VInha, VP Matti N, VP Heikki H, VP Jennifer P, VP Merja P, VP Carla M, VP Shirley C, VP Sanna A, VP Petteri V, VP Juhani M, VP Veikko VInha, VP Laila Lumme, VP Matti N, VP Selma S, VP Petteri V, VP Sirkka S, VP Jennifer P
- 130 VP Terttu S, VP Selma S, VP Max S, VP Juhani M, VP Jennifer P
- 133 VP Kerttu V., VP Juhani M, VP Merja P, VP Sanna A, VP Veikko VInha, VP Laila Lumme, VP Heikki H, VP Matti N, VP Pekka Oikarinen, VP Pekka Oikarinen, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin
- 134 VP Terttu S, VP Kauko R
- 135 VP Juhani M, VP Juhani M, VP Juhani M
- 139 VP Terttu S, VP Selma S, VP Sanna A, VP Sanna A, VP Sirkka S, VP Max S, VP Veikko VInha
- 152 VP Sanna A, VP Sanna A
- 156 VP Terttu S, VP Terttu S, VP Kauko R, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen
- 162 VP Kauko R
- 163 VP Selma S, VP Kauko R, VP Terttu S
- 165 VP Terttu S
- 166 VP Kauko R
- 168 VP Terttu S, VP George F
- 172 VP Terttu S, VP Selma S, VP Petteri V, VP Petteri V

(continues)

## Appendix 5. (continued)

- 174 VP Terttu S, VP Kauko R, VP Heikki H, VP Veikko Vinha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Shirley C
- 184 VP Terttu S
- 194 VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N
- 204 VP Terttu S, VP Kauko R, VP Bertta Berlin, VP Kauko R, VP Sanna A
- 206 VP Kerttu V.
- 208 VP Pekka Oikarinen, VP Terttu S, VP Kauko R, VP Kerttu V.
- 211 VP Petteri V, VP Selma S
- 215 VP Terttu S, VP Kauko R, VP Sanna A, VP Sanna A, VP George F, VP Max S, VP Sanna A, VP Heikki H, VP Max S, VP Max S, VP Heikki H, VP Max S, VP Jennifer P, VP Pekka Oikarinen
- 216 VP Terttu S, VP Sanna A
- 220 VP Sanna A
- 221 VP Terttu S, VP Max S, VP Jennifer P, VP Terttu S
- 225 VP Kauko R, VP Terttu S
- 226 VP Sanna A, VP Tuomas Tammi, VP Veikko Vinha, VP Laila Lumme, VP Bertta Berlin, VP Pekka Oikarinen
- 230 VP Selma S, VP Matti N, VP Matti N, VP Kerttu V., VP Sanna A, VP Veikko Vinha, VP Sanna A, VP Pekka Oikarinen, VP Veikko Vinha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP George F, VP Sanna A, VP Juhani M
- 232 VP Selma S, VP Selma S, VP Veikko Vinha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Sirkka S, VP Terttu S, VP Kauko R, VP Jennifer P
- 241 VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S
- 242 VP Terttu S, VP Kauko R, VP Terttu S, VP Veikko Vinha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Jennifer P, VP Terttu S, VP Sanna A, VP Shirley C, VP Shirley C, VP Kauko R, VP Petteri V, VP Kerttu V., VP Selma S, VP Max S, VP Juhani M
- 246 VP Terttu S, VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M

(continues)

## Appendix 5. (continued)

- 259 VP Terttu S, VP Kauko R, VP Selma S, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Shirley C, VP Max S, VP Juhani M, VP Petteri V, VP Selma S, VP Matti N, VP Kerttu V., VP Kauko R, VP Terttu S
- 261 VP Laila Lumme, VP Max S
- 263 VP Terttu S
- 265 VP Kerttu V., VP Petteri V, VP Petteri V, VP Selma S, VP Selma S, VP Kauko R, VP Jennifer P, VP Sirkka S, VP Bertta Berlin, VP Laila Lumme, VP Veikko VInha, VP Pekka Oikarinen
- 269 VP Kauko R, VP Terttu S, VP Terttu S, VP Matti N, VP Matti N, VP Selma S, VP Petteri V, VP Max S, VP Sanna A, VP Shirley C, VP Sanna A, VP Carla M, VP George F, VP Merja P, VP George F, VP Heikki H, VP Sirkka S, VP Petteri V, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Pekka Oikarinen, VP Bertta Berlin, VP Tuomas Tammi, VP Tuomas Tammi, VP Laila Lumme, VP Veikko VInha
- 272 VP Kerttu V.
- 275 VP Kerttu V.
- 281 VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen
- 283 VP Sirkka S
- 288 VP Kerttu V., VP Kerttu V., VP Selma S
- 293 VP Petteri V, VP Petteri V, VP Juhani M
- 295 VP Terttu S, VP Kauko R, VP Kerttu V., VP Pekka Oikarinen, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Heikki H, VP Sirkka S, VP Jennifer P
- 298 VP Terttu S, VP Terttu S
- 315 VP Matti N, VP Terttu S, VP Terttu S
- 318 VP Selma S, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Bertta Berlin, VP Pekka Oikarinen, VP Petteri V, VP Jennifer P
- 321 VP Terttu S
- 322 VP Amalia V

(continues)

## Appendix 5. (continued)

- 323 VP Kauko R, VP Kerttu V., VP Sanna A, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Merja P, VP Heikki H, VP Heikki H, VP Sirkka S, VP Kauko R, VP Kauko R, VP Sanna A, VP Shirley C, VP Carla M
- 324 VP Petteri V
- 325 VP George F, VP Heikki H, VP Sirkka S, VP Jennifer P, VP George F, VP Heikki H
- 328 VP Petteri V
-

## Appendix 6. (continued)

**Table A6.1.** Complete table of virtual patient sessions taking place in 5 hours.

User_id	Virtual Patient Session
4	[VP Kauko R], [VP Kauko R, VP Kerttu V.]
5	[VP Max S], [VP Sanna A]
9	[VP Selma S]
14	[VP Terttu S, VP Kauko R], [VP Petteri V, VP Matti N], [VP Sanna A, VP George F, VP Merja P], [VP Max S], [VP Veikko VInha]
17	[VP Terttu S, VP Kauko R]
21	[VP Terttu S, VP Kauko R], [VP Kerttu V.], [VP Matti N, VP Selma S], [VP Petteri V], [VP Merja P, VP George F, VP Sanna A, VP Max S, VP Juhani M], [VP Jennifer P, VP Sirkka S]
22	[VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Terttu S], [VP Terttu S, VP Kauko R, VP Kerttu V., VP Kerttu V., VP Matti N, VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Shirley C, VP Sanna A, VP Laila Lumme]
24	[VP Petteri V, VP Terttu S, VP Terttu S, VP Laila Lumme, VP Terttu S, VP Veikko VInha, VP Terttu S, VP Laila Lumme, VP Terttu S, VP Tuomas Tammi, VP Terttu S, VP Bertta Berlin, VP Terttu S, VP Pekka Oikarinen, VP Pekka Oikarinen, VP Terttu S, VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sirkka S, VP Sanna A, VP Shirley C, VP Carla M, VP Shirley C, VP Carla M, VP George F, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Heikki H, VP Sirkka S, VP Heikki H, VP Heikki H, VP Jennifer P]
34	[VP Terttu S], [VP Max S, VP Juhani M]
46	[VP George F], [VP Sanna A, VP Sanna A, VP Shirley C, VP Carla M, VP Jennifer P], [VP Merja P, VP Heikki H, VP Veikko VInha, VP Laila Lumme, VP Veikko VInha, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Terttu S]
50	[VP Sanna A], [VP Shirley C, VP Carla M, VP George F], [VP Selma S], [VP Kerttu V., VP Merja P, VP Heikki H, VP Sirkka S, VP Jennifer P]

(continues)

## Appendix 6. (continued)

- 58 [VP Terttu S, VP Kauko R], [VP Terttu S, VP Kauko R, VP Kerttu V., VP Terttu S, VP Matti N, VP Terttu S], [VP Sanna A, VP Veikko VInha, VP Terttu S, VP Shirley C, VP Sanna A, VP Carla M, VP George F, VP Carla M, VP Merja P, VP Heikki H, VP Sirkka S, VP Jennifer P, VP Petteri V, VP Juhani M, VP Max S], [VP Pekka Oikarinen, VP Bertta Berlin, VP Tuomas Tammi, VP Laila Lumme]
- 66 [VP Max S, VP Juhani M, VP Terttu S]
- 71 [VP Terttu S, VP Kauko R]
- 73 [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Heikki H], [VP Heikki H, VP Heikki H, VP Sirkka S, VP Jennifer P]
- 77 [VP Kauko R], [VP Heikki H], [VP Veikko VInha, VP Terttu S, VP George F]
- 79 [VP Tuomas Tammi]
- 81 [VP Kerttu V., VP Kerttu V., VP Terttu S, VP Kauko R]
- 84 [VP Petteri V], [VP Juhani M, VP Max S], [VP Max S]
- 90 [VP Terttu S], [VP Kerttu V.]
- 95 [VP Heikki H, VP Sirkka S, VP Heikki H, VP Petteri V], [VP Veikko VInha, VP Bertta Berlin, VP Laila Lumme, VP Tuomas Tammi, VP Pekka Oikarinen]
- 97 [VP Selma S], [VP Terttu S, VP Sanna A, VP Terttu S]
- 106 [VP Max S, VP Sirkka S], [VP Max S]
- 107 [VP Matti N, VP Selma S, VP Terttu S, VP Terttu S]
- 108 [VP Kerttu V., VP Kerttu V., VP Kauko R]
- 109 [VP Matti N, VP Selma S]
- 110 [VP Terttu S, VP Kauko R], [VP Kerttu V.], [VP Matti N, VP Selma S], [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen], [VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP Heikki H, VP Sirkka S, VP Jennifer P]
- 111 [VP Terttu S, VP Sanna A, VP Shirley C, VP Merja P, VP Veikko VInha, VP Petteri V, VP Terttu S], [VP Tuomas Tammi, VP Bertta Berlin, VP Veikko VInha, VP Laila Lumme, VP Pekka Oikarinen]

(continues)

## Appendix 6. (continued)

- 112 [VP Juhani M, VP Juhani M]  
113 [VP Kauko R], [VP Sanna A], [VP Merja P, VP Shirley C]  
114 [VP Kauko R, VP Terttu S], [VP Kerttu V.], [VP Tuomas Tammi, VP Laila Lumme, VP Veikko VInha, VP Bertta Berlin, VP Bertta Berlin, VP Pekka Oikarinen], [VP Sanna A, VP Shirley C], [VP Merja P, VP Jennifer P, VP Sirkka S, VP Heikki H, VP Shirley C, VP Carla M]  
115 [VP Tuomas Tammi]  
128 [VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S], [VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A, VP Shirley C, VP Carla M, VP Merja P, VP Max S, VP Carla M, VP Veikko VInha, VP Terttu S, VP Heikki H, VP Veikko VInha, VP Matti N], [VP Heikki H, VP Jennifer P, VP Merja P, VP Carla M, VP Shirley C, VP Sanna A, VP Petteri V, VP Juhani M], [VP Veikko VInha, VP Laila Lumme, VP Matti N, VP Selma S, VP Petteri V, VP Sirkka S, VP Jennifer P]  
130 [VP Terttu S, VP Selma S, VP Max S, VP Juhani M, VP Jennifer P]  
133 [VP Kerttu V.], [VP Juhani M, VP Merja P, VP Sanna A, VP Veikko VInha, VP Laila Lumme, VP Heikki H, VP Matti N, VP Pekka Oikarinen, VP Pekka Oikarinen, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin]  
134 [VP Terttu S], [VP Kauko R]  
135 [VP Juhani M, VP Juhani M, VP Juhani M]  
139 [VP Terttu S], [VP Selma S], [VP Sanna A, VP Sanna A, VP Sirkka S, VP Max S], [VP Veikko VInha]  
152 [VP Sanna A, VP Sanna A]  
156 [VP Terttu S], [VP Terttu S, VP Kauko R, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Max S, VP Sanna A], [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen]  
162 [VP Kauko R]  
163 [VP Selma S, VP Kauko R, VP Terttu S]  
165 [VP Terttu S]  
166 [VP Kauko R]  
168 [VP Terttu S], [VP George F]

(continues)



## Appendix 6. (continued)

- 172 [VP Terttu S], [VP Selma S], [VP Petteri V, VP Petteri V]
- 174 [VP Terttu S], [VP Kauko R, VP Heikki H, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen], [VP Shirley C]
- 184 [VP Terttu S]
- 194 [VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N]
- 204 [VP Terttu S], [VP Kauko R], [VP Bertta Berlin, VP Kauko R, VP Sanna A]
- 206 [VP Kerttu V.]
- 208 [VP Pekka Oikarinen, VP Terttu S, VP Kauko R, VP Kerttu V.]
- 211 [VP Petteri V], [VP Selma S]
- 215 [VP Terttu S, VP Kauko R, VP Sanna A], [VP Sanna A, VP George F, VP Max S], [VP Sanna A], [VP Heikki H, VP Max S, VP Max S, VP Heikki H, VP Max S, VP Jennifer P, VP Pekka Oikarinen]
- 216 [VP Terttu S], [VP Sanna A]
- 220 [VP Sanna A]
- 221 [VP Terttu S], [VP Max S, VP Jennifer P], [VP Terttu S]
- 225 [VP Kauko R, VP Terttu S]
- 226 [VP Sanna A, VP Tuomas Tammi, VP Veikko VInha, VP Laila Lumme, VP Bertta Berlin, VP Pekka Oikarinen]
- 230 [VP Selma S, VP Matti N, VP Matti N, VP Kerttu V.], [VP Sanna A], [VP Veikko VInha, VP Sanna A], [VP Pekka Oikarinen, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin], [VP George F, VP Sanna A, VP Juhani M]
- 232 [VP Selma S], [VP Selma S], [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Shirley C, VP Carla M, VP George F, VP Sirkka S, VP Terttu S, VP Kauko R], [VP Jennifer P]
- 241 [VP Terttu S, VP Kauko R], [VP Kerttu V., VP Matti N, VP Selma S]

(continues)

## Appendix 6. (continued)

- 242 [VP Terttu S], [VP Kauko R, VP Terttu S], [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Jennifer P], [VP Terttu S, VP Sanna A, VP Shirley C, VP Shirley C], [VP Kauko R, VP Petteri V], [VP Kerttu V., VP Selma S], [VP Max S, VP Juhani M]
- 246 [VP Terttu S, VP Terttu S, VP Kauko R, VP Kerttu V., VP Matti N, VP Selma S, VP Petteri V, VP Juhani M]
- 259 [VP Terttu S, VP Kauko R], [VP Selma S], [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen, VP Sanna A, VP Shirley C, VP Max S, VP Juhani M, VP Petteri V, VP Selma S, VP Matti N, VP Kerttu V., VP Kauko R, VP Terttu S]
- 261 [VP Laila Lumme, VP Max S]
- 263 [VP Terttu S]
- 265 [VP Kerttu V.], [VP Petteri V, VP Petteri V, VP Selma S, VP Selma S, VP Kauko R], [VP Jennifer P, VP Sirkka S, VP Bertta Berlin, VP Laila Lumme, VP Veikko VInha, VP Pekka Oikarinen]
- 269 [VP Kauko R, VP Terttu S, VP Terttu S], [VP Matti N, VP Matti N, VP Selma S], [VP Petteri V], [VP Max S, VP Sanna A, VP Shirley C, VP Sanna A, VP Carla M, VP George F, VP Merja P, VP George F, VP Heikki H, VP Sirkka S, VP Petteri V], [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi], [VP Pekka Oikarinen, VP Bertta Berlin, VP Tuomas Tammi, VP Tuomas Tammi, VP Laila Lumme, VP Veikko VInha]
- 272 [VP Kerttu V.]
- 275 [VP Kerttu V.]
- 281 [VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi, VP Bertta Berlin, VP Pekka Oikarinen]
- 283 [VP Sirkka S]
- 288 [VP Kerttu V., VP Kerttu V.], [VP Selma S]
- 293 [VP Petteri V], [VP Petteri V, VP Juhani M]

- 295 [VP Terttu S, VP Kauko R, VP Kerttu V.], [VP Pekka Oikari-  
nen, VP Veikko VInha, VP Laila Lumme, VP Tuomas Tammi,  
VP Bertta Berlin, VP Terttu S, VP Kauko R, VP Kerttu V., VP  
Matti N, VP Selma S, VP Petteri V, VP Juhani M, VP Sanna  
A, VP Shirley C, VP Carla M, VP George F, VP Merja P, VP  
Heikki H, VP Sirkka S, VP Jennifer P]
- 298 [VP Terttu S], [VP Terttu S]
- 315 [VP Matti N], [VP Terttu S, VP Terttu S]
- 318 [VP Selma S], [VP Veikko VInha, VP Laila Lumme, VP Tuo-  
mas Tammi, VP Bertta Berlin, VP Bertta Berlin, VP Pekka  
Oikarinen, VP Petteri V, VP Jennifer P]
- 321 [VP Terttu S]
- 322 [VP Amalia V]
- 323 [VP Kauko R], [VP Kerttu V.], [VP Sanna A, VP Sanna A, VP  
Shirley C, VP Carla M, VP George F, VP Merja P, VP Merja P,  
VP Heikki H, VP Heikki H, VP Sirkka S], [VP Kauko R, VP  
Kauko R, VP Sanna A, VP Shirley C, VP Carla M]
- 324 [VP Petteri V]
- 325 [VP George F, VP Heikki H, VP Sirkka S, VP Jennifer P, VP  
George F, VP Heikki H]
- 328 [VP Petteri V]
-