

# COSC1165/1167 Intelligent Web Systems

## Assignment 1

### Recommender System

### Due end of week 6

You may complete this assignment in groups of two or on your own.

#### Plagiarism

All assignments will be checked with plagiarism-detection software; any student found to have plagiarised will be subject to disciplinary action as described in the course guide. Plagiarism includes submitting code that is not your own or submitting text that is not your own. Submitting one comment in your code or a sentence from someone else's report is plagiarism, and plagiarism includes submitting work from previous years. Allowing others to copy your work is also plagiarism. All plagiarism will be penalised; there are no exceptions and no excuses. For further information, please see: <http://www.cs.rmit.edu.au/students/integrity/>  
You have been warned.

#### Implementation

You are to develop a recommender system for course selection. The recommender system keeps historical data for future cases. The format of such data is shown below:

**course\_id:** a *unique* numeric identifier for the course.  
**lecturer\_id:** a numeric identifier for the lecturer. (the same lecturer can teach multiple courses).  
**tutor\_id:** a numeric identifier for the tutor. (the same tutor can teach multiple courses).  
**core/elective:** a binary value indicating if the course is a core or elective. 0 means core, 1 means elective.  
**student\_1:** a score in the range [1..5] that the student\_1 gives the particular course. (Integer only)  
**student\_2:** a score in the range [1..5] that the student\_2 gives the particular course. (Integer only)

You can assume that students only give scores to the courses that they have completed.

An example dataset. Note there are a total of 10 students here.

course_id	lecturer_id	tutor_id	core/elective	student_1	...	student_10
1	1	1	0	4	...	2
2	1	2	0	4	...	2
3	2	1	1	2	...	3

In addition, the recommender will keep track of the following details about a student:

**student\_id:** a unique numeric identifier for the student. Note that a student with the student\_id 1 corresponds with the score given by student\_1.

**pg/ug:** a binary value indicating if the student is an undergrad or elective. 0 means postgrad, 1 means undergrad.

**m/f:** a binary value indicating if the student is male or female. 0 means male, 1 means female.

**int/local:** a binary value indicating if the student is international or local. 0 means international, 1 means local

**GPA:** a grade point average of the student in the range [1...4]. (integer only, 1,2,3,4, no other values.)

An example dataset. Again note that there are 10 students.

student_id	pg/ug	m/f	int/local	GPA
1	0	1	0	4
2	1	0	1	3
3	1	1	1	3
...	...	...	...	...
10	1	0	0	2

When provided with a sample student, the system will generate a list of recommended courses and the value used to rank the selection.

As part of your submission, you should briefly discuss your implementation, in particular, the algorithms you used, how the algorithms work and reasons for such choices. Also you should discuss your results with the sample data. Examples where the algorithm makes good recommendations and examples where the algorithm makes poor recommendations (and why?)

Your report should be in plain text format and no more than 1000 words.

A recommendation would include at most three courses in the order of preference, e.g. *Recommending Course 1, 5, 2.*

A sample HTML interface if you wish to implement in PHP/ASP/etc:  
(<http://goanna.cs.rmit.edu.au/~hali/iws.html>)

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://goanna.c.../~hali/iws.html`. The page content includes a 'Welcome' heading and a 'Tell me about yourself' section. This section contains several form elements: radio buttons for 'Postgrad' (selected) and 'Undegrad', 'International' and 'Local' (selected), 'Male' (selected) and 'Female', and a 'GPA' text input with the value '3'. Below these are four course selection columns. Each column has a 'Course' dropdown, 'Lecturer' and 'Tutor' text inputs, and radio buttons for 'Core' and 'Elective'. Course 1 has dropdowns set to '1' and '2'. Course 2 has dropdowns set to '5' and '1'. Course 3 and Course 4 have dropdowns set to 'NOT COMPLETED'. A 'Submit' button is located below the course selection area. At the bottom of the form is a 'Batch Testing csv file' label and an 'Upload' button.

You can assume that the student who uses this site has maximum 4 courses completed. It is also possible that they may have not done any courses at this time.

The sample record data and test cases will be released on the blackboard.

### **Bonus task**

For students that complete the initial implementation there will be an optional bonus task. This will involve keeping track of prerequisites and ensuring that the recommended subjects are not below the level of completed courses. Eg. A student that has completed "Programming 2" would not be recommended to do Programming 1 or "Introduction to Programming" as these would be pre-requisites.

### **Submission**

Assignments can be completed in any language as long as they can be tested on RMIT computers (labs or yallara). Other than the source code and the report, group submissions should also include a statement about the contribution of each member, either in the report or in a separate file.

Submission will be via Weblearn at the end of week 6 (Friday 13<sup>th</sup> April at 11:59pm)