International Journal of Engineering & Technology, 7 (3.15) (2018) 110-113

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Personalized Recommender System for Calculus using Content-Based Filtering Approach

Noor Latiffah Adam*, Muhammad Alif Zulkafli, Shaharuddin Cik Soh, Nor Ashikin Mohamad Kamal, Nordin Abu Bakar

Faculty of Computer & Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Malaysia
**Corresponding author E-mail: latiffah@tmsk.uitm.edu.my*

Abstract

In this millennial age, Internet is becoming essential to human kind. Along with the growth of Internet users, information is also becoming huge and starting to cause difficulties to find the relevant contents. Thus, the recommender system was introduced. It helps the user to suggest the items based on the user's preferences. This system could help the students as Calculus is one of the tough subjects feared by most students. Credits given to the technology as many sources on the web can provide tutorials, working examples and solutions on the subjects. However, there are too many of them. Students had to make a few selections, which one can fulfil their needs of specific calculus topics. The personalized recommender system developed was a content-based filtering recommender system with its own scraping engine to collect the sources from the Internet which focuses on the basic Calculus topics. The system and engine were constructed by using Flask framework together with its relevant libraries.

Keywords: *recommender system; calculus; content-based filtering.*

1. Introduction

The recommender system is popular and commonly used to help the Internet users by suggesting any related items according to the user's preferences. Thousands of applications that are based on recommendation systems were developed to solve the current problems in various areas such as e-commerce, education, and entertainment (1). Furthermore, the recommender system is not only suggesting relevant sets of the item, but it also pays attention to its precision, originality, and security of the data (2). Usually, the term '*item*' is often used to describe the contents in the recommender system. The recommender systems that are currently and widely used by worldwide users are *Flipboard, Netflix, Spotify, YouTube,* and *Amazon*. *Netflix, Spotify,* and *YouTube* are specialized in the entertainment area while *Amazon* is specialized in the e-commerce area.

Education is one of the fields that are popular when it comes to implementing the recommender system application. It is because learning is a process of getting and increasing one's knowledge through observation, study, and lessons of a person. Learning can be academic knowledge or non-academic knowledge. According to Siemen's (3), most of the learning approaches for schooling environment are behaviorism, cognitivism, and constructivism. These approaches are popular before the impact of technology. Different learning approaches have many divergent methods. One of the approaches is to educate the learner to be more self-regulated in the learning process (4,5).

Throughout the systematic analysis, reading on journals and articles in the Internet and also comparison between the existing systems, there are three main approaches of the recommender system that are Collaborative-Filtering, Content-based Filtering and Hybrid approaches (6–8). As for comparison between these existing systems, there are many online tutorials on a recommender-based system exist on the Internet such as YouTube, Khan Academy and Flipboard. The systems were built using the recommender system approach.

We are interested to look at the mathematic subject specifically calculus. The problem that the students faced in learning calculus are the classical closed classroom learning approach limits the personalized learning scenario for every one of them (9) and the lecturers faced difficulties in finding a suitable e-learning system that meets their needs (10). Some students are shy to share their problems with their lecturers during class because, they are afraid that their colleagues will laugh at them. Moreover, they cannot adapt the way their lecturers' approach in the class. Some lecturers are very firm and not student-friendly during the class session and make the students too afraid to approach them. Thus, students looked for online references, which could offer any personalized (9) learning material as their alternative solution to understand what they had learned in the class. This project offers one of the alternatives for these kinds of students. The system suggested is a recommender system that focuses on suggesting any suitable and relevant websites based on the basic calculus topics. The preliminary study of this project can be found in (11).

2. Research Method

The research framework for this project has been discussed in (11). User acceptance test has been carried out and reported in the discussion section. The test is essential to check its functionality and accuracy according to user preference. The design phase outline is shown in Figure 1. The following subsection will characterize the design phase of the project.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Fig. 1: Design phase flow

2.1. Project Design Environment

The system was developed using the Flask framework. Flask framework is a Python based programming language that focuses on web application development. This framework implements the *jinga2* template to allow the python code to be written in any html files. Furthermore, the system has used the Model-View-Controller (MVC) approach to make a better system design and maintenance.

2.2. Database Design

The system updates its content according to its engine which is responsible in data extraction from the website by using the web scraping and crawling tools. To make the system more dynamic, a database is required to make it easier to store and to retrieve the data. Thus, the relational database is a suitable database since it does not handle any complex data structure such as image, videos or other multimedia contents. For this section, a data dictionary was used to explain each table in the system's database.

2.3. Data Extraction Engine Design

Data extraction algorithm has been used to extract the informations from the targeted website. Filtration is the main process in this engine. There are two main processes of filtration in this engine, particularly, the similarity link and the classification web link filtration.

The engine was developed using the existing library in the Python programming language and the most suitable library to use is BeautifulSoup. BeautifulSoup library allows any requested website link to be read in the HTML parser format, which the library engine will read all the HTML tags and its properties from the targeted website. Thus, the extraction of the information can be executed by finding the HTML tags and attributes that should be extracted.

Figure 2 shows the pseudocode of the data extraction engine. The engine will use the topics and its keywords according to the data from the table Topic and Keywords, respectively. Then, the engine will concatenate the topic altogether with its keywords as the query to obtain the link results from it. Once the results retrieved, the filtration of the similarity links can be executed by checking the current links. If it is similar with the previous links, the engine will relinquish it and proceed to the next process.

After the process is completed, the engine will filter it once again by classifying the link based on their content. One thing to note that the files on the Internet is a multimedia content which means it's not only a content text, it may also contain other than that such as videos and images. Thus, the second process is responsible for classifying them based on the type of content. As an example, the url: www.exampleWebsite.com/sampleFile.pdf is a website that shows the pdf file content once it was clicked because of its extension is a pdf format. The second process is also responsible to read all the extensions from the filtered links and will only select the HTML or HTM extensions. Finally, the link that had successfully filtered by both procedures will be extracted and saved to the text file format.

Algorithm: Data extraction engine pseudocode.
Procedure: Extracted web links filtration.
Input: Topics and its keywords
Output: Write web contents into text file.

```
1.  Read topic content and its keyword from database
2.  Links = {}
3.  Get all URL links from web scraping tools
4.  For each link in URL links do
5.      If Links is empty
6.          Append into Links
7.      Else
8.          If link is similar with link in previous index then
9.              isSimilar = True
10.         Else
11.             isSimilar = False
12.         End If
13.         If isSimilar == True then
14.             Append into Links
15.     End If
16. End For
17. WebLink = {}
18. For each item in Links do
19.         If item contain "youtube" do
20.             BREAK
21.         Else
22.             Append into webLink
23.         End If
24. End For
25. For each item in webLink do
26.         Extract information related item
27.         Append into text file
28. End For
```

Fig. 2: Data extraction engine pseudocode

3. Results and Analysis

This section describes the results and the findings throughout the project's development process. There are two main issues that will be discussed in this section: the overall system functionalities and the data extraction engine results.

3.1. Login User Interface

The *login* page is the first page that users will notice when they are accessing the system. The *login* page's purpose is to authenticate and validate the user either his/her entity exists in the system or not. If it exists, the system will be redirected to index.html page or else, it will prompt an error message. Table 1 shows the components or the input, output and the process in this page.

Table 1: *Login* user interface attributes

Input	Process	Output
1. Email Field 2. Password Field	Authenticate the user entity either the user is exist or not in the system when clicked "Sign in" button.	1. If user's entity exist - Proceed to *index.html* page 2. If user's entity not exist - Prompt error message at *login.html* page

3.2. Signup User Interface

The *signup* page is an optional page that users may see and use if they are new to the system. Table 2 shows the input, the output and the process of this page.

Table 2: *Signup* user interface

Input	Process	Output
1. Email 2. Password 3. Confirmation Password	Create an entity for new user by adding user's information into database when user click "Sign up" button.	1. If password and confirmation password is matched - Proceed to *interest.html*

		page 2. If password and confirmation password is not matched Prompt error message at *signup.html* page

3.3. User Interest Interface

The *interest* page is the second page that users will see after they finished inserting their emails and passwords from the *signup.html* page. This page is responsible for displaying the topics saved from the database and captured all the interests chosen by the users as they have their preferences. This could help the system to suggest the web content based on the topic chosen by the user. Table 3 shows the input, output and the process of this page.

Table 3: User's interest interface

Input	Process	Output
Checked radio button	1. Read all topic registered from database. 2. Captured all checked radio button from user 3. Record user's preferences into database interest table	Add into interest table based on userNo and topicNo

3.4. Index Interface

The *index* page is the main page of this system. This page is responsible to display only the contents based on the user's preferences recorded from *interest.html* page earlier. Table 4 shows the input, output, and the process of this page.

Table 4: *Index* interface

Input	Process	Output
Header hyperlink	Redirect to link stored in href atribute in <a> html tag	Redirect and display the website link in the new tab
'Find my interest' button	Allowing users to change and update their current preferences	Redirect to *modifiedInterest.html* page
Topic hyperlink	System will read topic's href attribute value in <a> and point to <section> which have similar value.	Navigate users to selected choice of topic's web content
'Logout' button	Destroying session once user choose to log out from the system	Redirect to *login.html* page

3.5. Data Extraction Engine

The data extraction engine is the main process and the backbone of this recommender engine. This engine contains several processes just to make sure that the data that had been extracted from the Internet is clean and well structured. Table 5 shows the input, the output and the process of the data extraction engine. Figure 3 shows the code snippet of the output data to the text file from the data extraction engine and the actual written output from the text file format.

Table 5: Data extraction engine input, output and process table.

Input	Process	Output
Target website link	1. Capture all <a> html tag from target link. 2. Similarities filtering process for each link capture 3. Classification filtering process for each link	1. Separate the attributes of web content by ';' (semicolon) 2. Store into text file

```
file = open("Integration of Rational Function.txt", "a")
for link in webLink:
    page = requests.get(link.getLink())
    soup = BeautifulSoup(page.content, "html.parser")
    contents = soup.find("p")
    scores = soup.findAll("p")

    keywordCounter = 0
    for score in scores:
        if "Integration" or " Rational Function" in score:
            keywordCounter = keywordCounter + 1

    file.writelines(link.getTitle() + ";" + link.getLink() + ";" +
str(keywordCounter) + ";" + str(contents.string)+"\n")

file.close()
```

```
Integration of Rational func-
tions;http://www.sosmath.com/calculus/integration/rati
onal/rational.html;25;None

Integration of Rational Func-
tions;https://www.math24.net/integration-rational-
functions/;20;A rational function \(\large\frac{{P\left( x
\right)}}{{Q\left( x \right)}}\normalsize,\) where \({P\left( x
\right)}\) and \({Q\left( x \right)}\) are both polynomials,
can be integrated in four steps

Integration with partial fractions (practice) |
Khan Acade-
my;https://www.khanacademy.org/math/calculus-
home/integration-techniques-calc/partial-fraction-
expansion-calc/e/integration-of-rational-functions-by-
division-and-partial-fractions;3;If you're seeing this mes-
sage, it means we're having trouble loading external re-
sources on our website.
```

Fig. 3: Output of structured data written in text file format.

4. Discussion

The system itself contains five html files format that are the *login*, *signup*, *interest*, *modifiedInterest* and *index*. The *login* is responsible to authenticate the user's entity from the database. If the user is new to the system, he/she can register his/her entity and preferences into the database. Once it has been registered or passed the authentication and the verification process stage, session will be created, and the user may access the index.html, which is the main page of the system. Once the user has logout from the system, his/her session will be destroyed as one of the security's mechanism. Data extraction engine involved the processes of storing the extracted data from the targeted website into the text file format.
Usability testing was used to measure the system effectiveness among its user, which were students whose taking Calculus subject. Seven students were volunteered to test the system. This testing contains three questions that required tester to answer after they used the system. The test was conducted using Google Form, which after respondent response to all the questions, it will show summary of responses in bar chart figures.

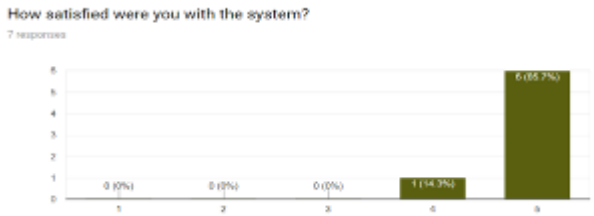


Fig. 4: Response summary of Question 1

Figure 4 shows the bar chart of respondent's responses of first question in the usability test. According to the results, majority of the respondents were very satisfied with the system. User experi-

ence was considered during system design and development. Easy to use and navigate is one of the user experience that was implemented in this system. Too much button and user input make the system hard to use.



Fig. 5: Response summary of Question 2

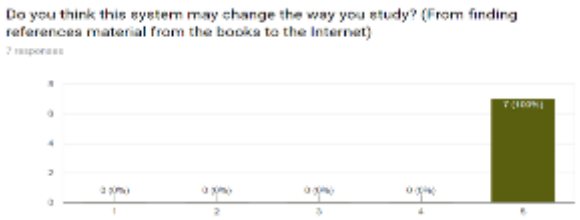


Fig. **Fig. 6:** Response summary of Question 3

Figure 5 and Figure 6 show the bar chart of respondent's responses of Question 2 and Question 3 in the usability test, respectively. In Question 2, all the respondents were agreed with the web content suggested to them based on their preferences as relevant reference material and helpful to them. In Question 3, they also agreed that this system could change the way they study Calculus. Before using this system, they were only exposed to lecturer notes and slides to study the topics that they wanted to revise. Some of the student did not understand the notes and slides provided by their respective lecturer. Thus, by using this system, they have an option to choose which website that could provide them with better explanation that suites them.

5. Conclusion

Since this is the first version of its kind, several deficiencies have been identified, namely, some websites cannot be crawled due to the site's security. As for this project, we will only cover the basic Calculus. The engine can be improvised to accommodate other subject of interest. The system may implement the collaborative filtering to increase the level of recommendation accuracy. Collaborative filtering contains a rating system, which could be used to rank the items of the system based on the user's preferences. For example, each of the web content may implement the 'like' function, which acts as the counter of the items. The higher the counter value, the higher the rank of the item gets. Furthermore, adding sharing function may allow user to share the content with others. This also may give an impact on the content and its rank.

Acknowledgement

This work is supported by UiTM Internal Research grants Scheme (600-IRMI/DANA 5/3/ARAS (019/2016) from the research Management Center (RMC), Univeristi Teknologi MARA, Shah Alam, Selangor. The appreciation also goes to Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor for giving a moral support in the production of this paper. **MyIPO references: LY2018002253**.

References

[1] Anusha UA, Biradar S. Recommender Systems : A Survey. Int J Latest Technol Eng Manag Appl Sci. 2016;5(1):42–5.

[2] Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender systems survey. Knowledge-Based Syst. Elsevier B.V.; 2013;46:109–32.

[3] Siemens G. A Learning Theory for the Digital Age. Int J Instr Technol Distance Learn. 2005;

[4] Adam NL, Alzahri FB, Cik Soh S, Abu Bakar N, Mohamad Kamal NA. Self-Regulated Learning and Online Learning: A Systematic Review. In: Advanced in Visual Informatics. Springer International Publishing AG; 2017. p. 143–54.

[5] Zimmerman BJ. Self-Regulated Learning and Academic Achievement: An Overview. Vol. 25, Educational Psychologist. 1990. p. 3–17.

[6] Asanov D. Algorithms and Methods in Recommender System. 2011.

[7] Thorat PB, Goudar RM, Barve S. Survey on Collaborative Filtering and Content-Based Recommending. Int J Comput Appl. 2015;110(4):31–6.

[8] Kanetkar S, Nayak A, Swamy S, Bhatia G. Web-based personalized hybrid book recommendation system. 2014 Int Conf Adv Eng Technol Res ICAETR 2014. 2014;0–4.

[9] Bourkoukou O, Bachari E El, Adnani M El. A Personalized E-Learning Based on Recommender System. Int J Learn Teach. 2016;2(2):99–103.

[10] Sommer T, Bach U, Richert A, Jeschke S. A web-based recommendation system for engineering education e-learning systems. CSEDU 2014 - Proc 6th Int Conf Comput Support Educ. 2014;1:367–73.

[11] Adam NL, Zulkafli MA, Soh SC, Ashikin N, Kamal M. Preliminary Study on Educational Recommender System. In: IEEE Conference on e-Learning, e-Management and e-Services (IC3e 2017). 2017.