ONLINE COURSE PERSONALIZED RECOMMENDATION SYSTEM

61431-FCS BSc (HONS) COMPUTING

19086235D
Jan 3, 2023

Project Supervisor: Prof. LI Qing

Abstract

Deciding on which course to take during undergraduate study is important to gain relevant knowledge of certain professions (game developer, etc.) and maintain satisfactory Grade Point Average (GPA). Steps needed to get a clear understanding of a course is a tedious process, hence the idea of introducing a personalized recommendation system. The aim of this project is to create a content-based filtering recommendation system to provide courses that suits to the users' preferences to help solve information overload issue and help reduce the steps needed to understand courses workloads, learning style, etc.

Acknowledgement

I would like to express my gratitude towards Prof. LI Qing, for supervising in which sections needed working and help me understand on what to write for this project.

Table of Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
UI	User Interface
UX	User Experience
CBF	Content-Based Filtering
CF	Collaborative Filtering
RecSys	Recommender System
IO	Information Overload
TF-IDF	Term Frequency – Inverse Document Frequency
NLP	Natural Language Processing
MAE	Mean Absolute Error

Chapter 1. Introduction

The following chapter introduces the background, motivation as well as objectives of the project, followed by the outline of the remaining part of the report.

1.1. Background and Motivation

Registering elective courses for undergraduate students is one of the most important processes in their study. Each students have different aspirations (game developer, software engineer, fintech specialist, data scientist, etc.). To achieve their aspirations, they must take relevant courses to gain required knowledge to excel in their respective fields. Neglecting this process may cause students to spend their study credits on irrelevant courses and may affect Grade Point Average (GPA) negatively due to low course satisfaction. Hence students need to plan this process carefully and they need to consider the aspects of each course (is the course relevant to students' study plan, difficulty, workload, teacher's teaching style, is the material up to date, etc.). Students who care about their future career prospects and maintaining higher grades have several steps that can be taken to help them conclude which course to register for. However, there are some pitfalls within these options, so the idea of each university having their own Recommendation System (RecSys) would be undeniably beneficial to both students and lecturers alike by recommending courses relevant to student's study plan (recommend game development-related classes and its pre-requisites for aspiring game developers).

The current situation and problems of known steps to the students' traditional decision-making process will be introduced as follows. Then the benefits of having a RecSys in higher-education institutes will be described. Continued with descriptions of several known techniques used in RecSys. Followed by the reason why Hybrid filtering method is picked for this project. After that, methodologies that will be taken for this project is shown and continued with the current progress of the project and wrapped up with the conclusion chapter.

When deciding on courses to register, Students may take several steps in considering whether an elective course is beneficial or not. They may ask their peers or academic advisors for advice on which courses to take based on their aspirations. However, there are several drawbacks to this approach as every student has different academic prowess and interests which could lead to unsatisfactory course experience based on the wrong information. Additionally, students who don't have acquaintances in the study field they are taking, especially international freshmen would have a hard time getting advice from their peers alongside the fact that the pandemic has made it harder for students to connect with their peers.

Students are advised by academic advisors to read the syllabus of each course to have a general idea of the course’s workload, etc. Despite that, this solution doesn’t provide insight on each lecturer’s teaching style. Chetty et al [1] claimed that styles of teaching affects student’s learning motivation which contributes to students’ academic performance. With that in mind, conflicting learning and teaching style may result in lower GPA. Furthermore, Schmitt et al [2] argued that students lacking the ability to seek for information are prone to experience Information Overload (IO), where the students’ mental capacity can’t process large amount of information when concluding whether a course is worth taking, hindering their decision-making process. This phenomenon happens when students gather data from several sources (peer advice, syllabus reading, etc.). This phenomenon happens when students synthesize information from several sources, reflect on each impact generated by different courses, and to conclude the most beneficial decision [3].

Some universities implemented ‘trial class’, where students can take a course during the beginning of each semester so students can have a first-hand experience of teacher’s teaching style and drop the course if students’ expectations are not met. However, this approach is impractical due to limitations in actions students can do to adjust their courses, leaving students unable to freely try out different electives.

Lastly, students are encouraged to use a combination of these three approaches to understand each courses’ learning and teaching aspects. Using this method offers a significant amount of course insight as each approach is used to cover up the other approaches’ weaknesses. Nevertheless, this process is time-consuming and is prone to cause IO to the students. Which is why the idea of having RecSys is introduced.

1.2. Benefits of using RecSys

Personalized recommendation services provided by RecSys reduces the issue of IO by recommending relevant course in regards to the students’ preferences [4]. With that in mind, RecSys may recommend courses relevant to their chosen interests and students don’t have to spend too much time on thinking and get on with their daily tasks. Furthermore, as user rates courses, lecturers teaching courses that have low rating number are encouraged to improve their course (modifying teaching style, difficulty, workload, material relevance, etc.) so that students can enjoy the course and their university life.

1.3. Project Objectives

There are 3 tasks in this project. For task 1, it is imperative to create a website that can gather user course interaction data (explicit data) and store them into a relational database for further processing. Moreover, this website should be able to send request and receive responses from the API server. Task 2 is to create an API server that can receive requests and send responses to the front-end website application. Task 3 is to create a hybrid RecSys consisting of Content-Based Filtering (CBF) RecSys and Collaborative Filtering (CF) RecSys and implement both RecSys into the API server.

Chapter 2. Literature Review

This chapter presents the conducted literature review for this project. It highlights several types of known personalized recommendation systems such as content-based filtering, collaborative filtering, and hybrid system. Alongside that, strengths and weaknesses associated with these systems are briefly explained. Lastly, an explanation containing why hybrid recommendation system is chosen will be given below.

2.1.1. Content-Based Filtering (CBF)

Wang et al [5] described this type of RecSys as a system that compares data regarding users searched and rated item. Concluding from the statement before, only the user data is used. Recommendations made by CBF RecSys relies heavily on the item's features, or courses' features (objectives, intended learning outcomes of the course) in this case to provide similar courses. To illustrate this, a figure explaining how CBF RecSys works in shown in Figure 2.1. below.

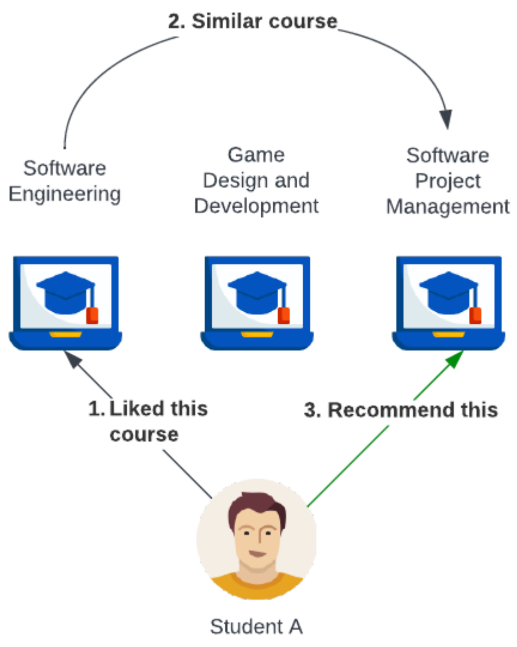

Figure 2.1. CBF recommender system example

Referring Figure 2.1. above, suppose student A took a Software Engineering course and rated the course positively, they will be recommended to take the Software Project Management course for the next semester due to a degree of similarity in the course's contents. One approach that can be used to determine the similarity of items is by using cosine distance approach. This approach has several steps needed which are, gathering data that describes the item, cleaning the data from stop words (a list of words that are frequent in any language which can be considered insignificant such as prepositions, pronouns, etc.), process the cleaned data into vectors using Term Frequency – Inverse Document Frequency (TF-IDF) technique which will then be used to find the cosine distance. With this approach, the resulting values will be ranging from 0 to 1, where the higher the value, the likelier an item is correlated to the item in question and vice versa. A strength from this type of RecSys is that there is no need for other users' data to start recommending personalized courses to the original user [6], [7]. Moreover, CBF RecSys provides transparency by explaining how the RecSys works [7]. On the other hand, this system's weaknesses are content overspecialization, where users only receives recommendations in regards to defined items in their user profiles and this type of RecSys' effectiveness relies heavily on available descriptive metadata [8]. Moreover, the effectiveness of this RecSys is limited by the person's knowledge when attributing descriptions to an item. However, this issue is not a case as this project uses course subject description forms provided by PolyU.

2.1.2. Collaborative Filtering (CF)

This type of RecSys generates recommendations by grouping users together and recommend them with items that have been rated by other users within the same group. Figure 2.2. below illustrates this type of RecSys in general.

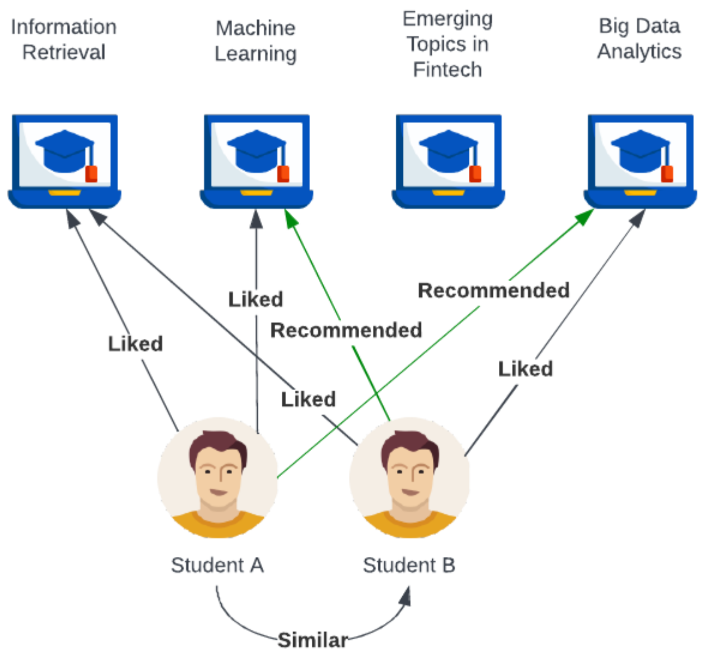

Figure 2.2. CF recommender system example

Referring to Figure 2.2. above, suppose that subjects liked by student A and student B overlaps, CF RecSys draws a relation that these students have similar preferences (ex. both student wants to be a machine learning engineer in the future) and thus, predicts that a course liked by student A will be liked by student B and vice versa. There are two main implementations as to how CF RecSys comes with these predictions, and they can be categorized into memory-based and model-based CF RecSys which will be explained briefly. For memory-based technique, it can be further divided into user-based and item-based collaborative filtering. User-based and item-based CF relies on grouping either users or items together to generate recommendations. On the other hand, model-based technique uses several algorithms such as matrix factorization (SVD, ALS, PLSA), clustering, etc. [8]. Such RecSys is commonly used to recommend items that is difficult to describe with metadata such as video, music, etc. [8]. It is inferred this RecSys combats the issue of CBF's content overspecialization and is not limited by a person's knowledge when attributing any item. Despite that, this approach brings up the cold-start problem, where such systems require large datasets on users interaction with the offered items first before it can generate accurate recommendations [7]. In conclusion, deploying CF RecSys is impractical for businesses with low number of recurring users because gathering the large dataset requires a long time. Furthermore, using such technique costs a lot of time due to computation growing linearly with number of users and items.

2.1.3. Hybrid Recommender System

Hybrid RecSys, as it names suggests, this system uses a combination of CBF and CF RecSys to provide recommendations with better accuracy than RecSys using only one filtering implementation [6], [7]. This hybrid technique offsets each technique's weaknesses by implementing the other's strengths. For example, it is known that CF method suffers from cold-start problem and in this case, recommendations generated by CBF RecSys can be used while gathering relevant dataset regarding user's interaction with certain items. When the system has enough data to build a reliable user profile, CF method will be implemented to stop from over-specialization issue brought up by CBF. Through combining both types of RecSys, users are given recommendations that are broad and accurate. However, the time needed grows as there are more than one RecSys that needs to be built.

2.2. Why Hybrid Recommender System

To reiterate, implementing a hybrid RecSys yields the most accuracy. The system can gather interaction data between a user and a course while recommending courses provided by CBF RecSys using subject description form. When enough data is gathered, recommendations generated by CF RecSys will be given alongside the initial CBF RecSys. This approach will be used for the project to provide diverse recommendations. For CBF RecSys, TF-IDF and cosine similarity method will be used to generate whether a course is correlated or not to another course. For CF RecSys, a model-based approach using a dimensionality reduction technique named Alternating Least Squares (ALS) will be used.

Chapter 3. Methodology

In this section, there are several important parts which spans through Data collection, RecSys development, Database design, Front-end/website application development, and Back-end/API server development to implement a website app that can recommend relevant courses. The following section will explain how each parts plays a role in this project which will be wrapped up by accuracy evaluation section.

3.1. Data Collection

For this project, there are two types of recommendation that is offered which is CBF and CF recommendations. These type of RecSys requires different data gathered and the following sub-sections will explain the steps needed to gather respective data.

3.1.1. Content-Based Filtering Data Collection

For CBF RecSys to start generating recommendations, PolyU course data must be gathered. Fortunately, these course data are available online as subject description forms. For example, computing subject description form's important features are objectives, intended learning outcomes, subject synopsis/indicative syllabus. However, it is important to note that different department presents descriptive data differently. Looking at Chinese Bilingual Studies (CBS) department's subject description forms, there is only two important features which means these features must be merged into one feature. Moreover, it is possible that some departments do not offer subject description forms online. For now, gathering subject description form will be conducted online. After gathering enough description data, the features within the data are merged as one which will then be inserted into the database in section 3.2.

3.1.2. Collaborative Filtering Data Collection

As stated above, this type of recommendation will only be given when enough user data is gathered. This kind of recommendations requires gathering several user's existing data such as courses that have been taken and ratings given by the user. For example, users that have been taking machine learning courses will be grouped with similar users and receive courses recommendations from the group. After basic functionalities of the front-end website have been implemented, a course rating feature will be added for data gathering uses. It is important to note for this project that data pre-processing is unnecessary.

3.2. Database Design

For this section, several essential tables like user, course, and rating are needed for this project to achieve the personalized recommendation feature. However, it is important to note that the database structure might change during this project's development lifecycle. an ER diagram shown in Figure 3.1. below helps reflect each table's relation to other tables.

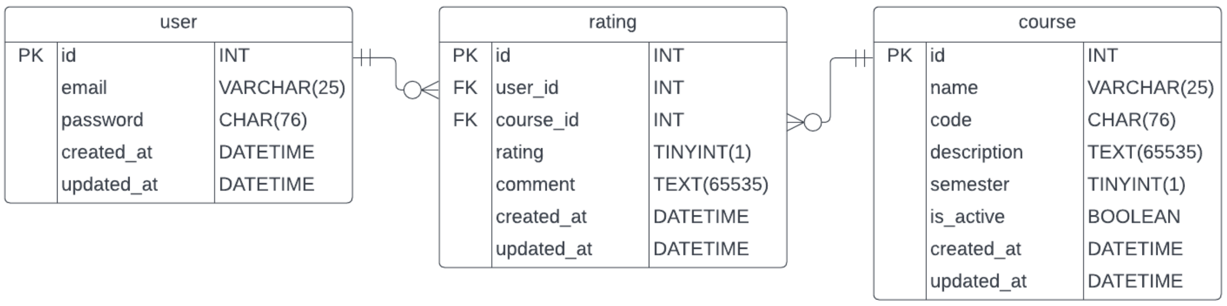

Figure 3.1. ER Diagram of the database

For usage with CBF RecSys, only the course table is used, specifically the description column. On the other hand, the rating table containing data about user-course interaction is used for CF RecSys. As of the writing of this report, the comment column will not be used.

3.3. RecSys

This hybrid RecSys will be built using Python programming language, which allows the leverage of several existing Python libraries such as **Pandas**, **Scikit-learn**, **PySpark**, and **NumPy** to generate recommendations from relevant data. The following sub-sections will explain the steps needed to build both RecSys.

3.3.1. Content-Based Filtering Recommender System

Taking the merged description data from the database, the data needs to be pre-processed before further processing. The data usually needs to be cleaned as there is a possibility of redundant data and to improve data quality [9]. To achieve this, **TfidfVectorizer** from **scikit-learn** library is used to clean the data from stops words, a list of unnecessary frequent words (to, the, as, etc.) resulting in data with less noise and convert the cleaned data into vectors that can be plotted. Lastly, values returned from cosine similarity method can be used to determine whether a course is related to a certain course.

3.3.2. Collaborative Filtering Recommender System

For this method, library **PySpark** is used to implement ALS algorithm onto the rating table from the database containing user-course interaction data. As fitting the model takes a long time, it is unfeasible to fit the model on demand because users will have to wait a long time before getting any recommendations. To solve this issue, model training will be scheduled daily at a certain time and the resulting model will be saved into the local filesystem of the API server for consequent use.

3.4. Front-end Development

This website will be the place where users interact with the project, where relevant course data are sent from the back end / API server to be visualized to the website in a manner to prevent IO. The following sections consists of website pages, login/register feature, localization, and why website is chosen.

3.4.1. Website Pages

The website will have several pages with its purpose briefly listed in Table 3.1 below. Some of the pages are only available when the user has logged in to the website and some pages will be optional due to time constraints (ex. Privacy policy statement page).

Table 3.1. Website pages and its purpose

Page Name	Purpose
Home	Serves as the first page users go into. Includes a section to introduce the user what this website is about.
Courses	Show a list of courses that the user is interested in. In this page, the user can search for certain courses using a provided search bar from the website.
Course info	Show course's information from the gathered subject description form and list other courses that is like this course through CBF. Furthermore, user can add the course into a list of courses they had taken and give a rating towards the course for CF recommendations.
Profile	Show courses that have been taken and their ratings by the user, where they can modify the ratings and remove a course from the taken course list if they had made mistakes.

Documents	Contains several capstone project submissions ranging from project outline to final report.
Privacy Policy (Optional)	Outlines how I, as the project developer uses the data only for this project, ensuring the user's private data.

Inferring from Table 3.1. above, the website will be able to query the API server and visualize the recommended course data from the API server's response to help the students' decision-making process. Furthermore, the website allows Create, Read, Update, Delete (CRUD) operations on the back-end server to ensure data sent to the users are up to date.

3.4.2. Login/Register Feature

A feature that hasn't been mentioned is the login/register feature as these will be shown in a modal pop-up shown in Figure 3.2. below. It is crucial to have a user account feature for this project because users will be able to rate courses they have taken, and these data will be taken into consideration for collaborative recommendations. To implement login/register feature, more research needs to be done on how it can be implemented due to security reasons. There are several available security methods that can be utilized for this project, but I will choose JSON Web Token (JWT) approach as it is quick and efficient to use for API authentication. Additionally, if a user who haven't logged into their account tries to see any course information, they will only be provided with recommendations generated by CBF method because the system requires the user to be logged into the website application to provide recommendations provided by CF method.

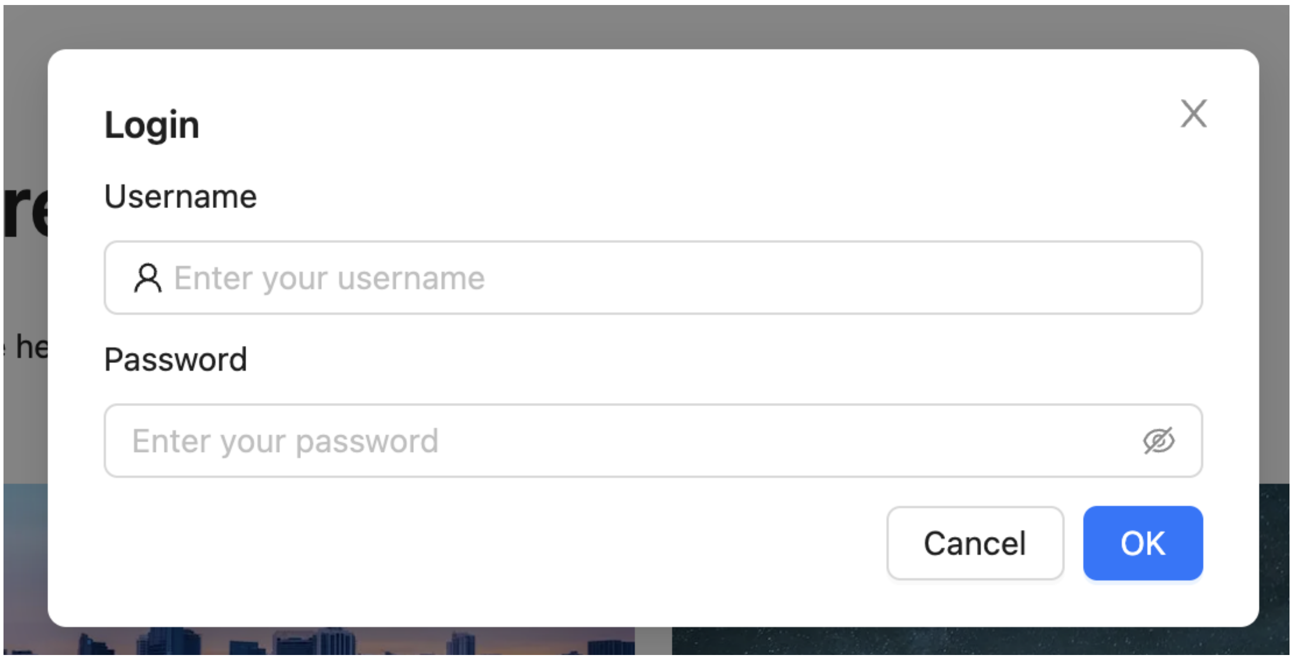

Figure 3.2. Login modal

3.4.3. Localization

An additional feature added to this website is to change the language of the website as PolyU's student community is diverse. Currently, there are several languages provided with the help of both DeepL, a neural machine translation service, and my native language knowledge of Indonesian such as English, Chinese (Simplified), Indonesian, Japanese. However, important course information will not be translated (ex. course name, objectives, intended learning objectives, and subject synopsis/indicative syllabus) to prevent mistranslations of the course description. Additionally, I might plan on asking for the help of other students to help with checking the correctness of translations generated by DeepL.

3.4.4. Why Website Application

Two main factors are considered before choosing a website application approach. A study conducted found that wireless traffic had seen an 73% increase within the span of two years from 2007 to 2009 [10], which signifies the importance of providing a website application that can be accessed by both smartphones and desktops over developing a mobile or desktop application that is not fully accessible by other devices and vice versa. Secondly, I'm familiar with website application development process using NextJS, a front-end JavaScript framework built on top of ReactJS that provides many out-of-the-box features such as hybrid static & server-side rendering, smart bundling, route prefetching, image optimization and thus improving the developer experience without unnecessary configurations. Because of this, development with NextJS will consume less time over using mobile or desktop application development technologies such as React Native or ElectronJS due to unfamiliarity.

3.5. Back-end Development

Django, a python web framework will be used for this project to provide personalized recommendations. Django is chosen as the RecSys is programmed in Python. This API server will be able to receive requests and send response to the front-end website. The following sub-section discusses about database and RecSys integration.

3.5.1. Database Integration

To fully integrate the database into the API server, there needs several endpoints that does CRUD operations. More research needs to be done on which database to use, ranging from MySQL, PostgreSQL, MariaDB, Oracle Database. However, for this project, a relational database will be used as there are tables that have relations with other tables.

3.5.2. RecSys Integration

Integrating the RecSys into the API server can be done last when basic API functionalities is finished. This is due to both RecSys, and API server programmed in Python language. However, more research needs to be done on scheduling model training jobs for the Django application.

3.6. Accuracy Test

Evaluating the accuracy of RecSys is important to ensure that generated recommendations are correct or not. In this report, two methods to test the RecSys accuracy is used. Clickthrough Rate (CTR), a measurement of how many clicks generated by users seeing an advertisement which is course in this case. The value of CTR can be computed by dividing the number of clicks with the number of impressions (number of recommended courses that have been shown). A good CTR is 5%. Mean Absolute Error (MAE) is another evaluation metric that can be used to measure the accuracy of the predictions generated by the RecSys. The closer the value is to 0, the more accurate the model is. Fortunately, **PySpark** provides an evaluator method that can evaluate the MAE value of trained model which helps in getting features within schedule.

Chapter 4. Progress

As several subject description forms have been gathered and DB structure have been designed, the next step is to connect the database to the API server followed with implementing use login and use registration feature between the front-end website and API server for future data collection purposes. After that, both CF and CBF RecSys will be implemented. A table is given below to help understand the list of tasks that needs to be implemented for this project.

Table 4.1. Project schedule

Date	Milestone	Status
October 2022	**Deliverables of Project Outline:** - Detailed Project Plan **Research:** - Currently known recommendation techniques - Technologies that can be used for this project (React, Django)	Done
November 2022	**Implementation**: - Prototype user interface of the application	Done
December 2022	**Research**: - Do in-depth analysis of other works **Implementation**: - Create a prototype front-end website for displaying courses - Finalize which recommendation techniques to use - Gather relevant data about PolyU computing courses for recommendation - DB structure design	Done
January 2023	**Deliverables of Interim Assessment**: - Interim report - Presentation video and PowerPoint file - Meeting notes with supervisor **Research**: - Connecting the website with the API server - User login and registration **Implementation**: - API server with basic functionalities - DB implementation and integration to the API server	Pending

	- Implement user login and registration functions to both website and API server - Manually input all gathered data about PolyU courses to the DB	
February 2023	**Implementation**: - Add recommendations using CBF to the API server based on each course' overview - Continue work on the website to gather user data for CF recommendations - Offer CF recommendations when enough user data is gathered during deployment	Pending
March 2023	**Implementation**: - Testing app by asking a sample of computing students to try the application - Optimize both website and API server if needed - Bug fixes if needed	Pending
April 2023	**Deliverables of Final Report**: - Final Report - Final Presentation with project demonstration	Pending

Chapter 5. Conclusion

In view of the difficulties of deciding which university courses to take, it is imperative to help reduce Information Overload issue by providing a web-based hybrid RecSys that offers personalized course recommendations. This project aims to remove the steps needed for students (ex. reading course syllabus, trying out trial classes, etc). to conclude whether a course is suitable for them. In the past semester, research on available types of RecSys has been conducted, such as Content-Based Filtering (CBF), Collaborative Filtering (CF) RecSys, and hybrid RecSys. CBF RecSys uses course description to generate “similar courses” recommendations with the help of text vectorization using Term Frequency – Inverse Document Frequency (TF-IDF) and calculating the angle distance between data vector points. CF RecSys, on the other hand, uses a dimensionality reduction technique called Alternating Least Squares (ALS) to generate recommendations from a group of users.

For this project, a prototype website application has been developed, however there are several things that needed progress such as API server development, Database integration with API server, and hybrid RecSys implementation. For reference, Table 4.1. is given above to understand the current progress of this project. It is scheduled that by April 2023 along with the submission of the final report, all the important features will be finished and available for public use (website link will be provided in the final report).

References

[1] N. D. S. Chetty *et al.*, “Learning Styles and Teaching Styles Determine Students’ Academic Performances.,” *Int. J. Eval. Res. Educ.*, vol. 8, no. 4, pp. 610–615, 2019.
[2] J. B. Schmitt, C. A. Debbelt, and F. M. Schneider, “Too much information? Predictors of information overload in the context of online news exposure,” *Inf. Commun. Soc.*, vol. 21, no. 8, pp. 1151–1167, 2018.
[3] D. Dean and C. Webb, “Recovering from information overload,” 2011.
[4] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: A survey,” *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015, doi: 10.1016/j.dss.2015.03.008.
[5] Y.-Y. Wang, A. Luse, A. M. Townsend, and B. E. Mennecke, “Understanding the moderating roles of types of recommender systems and products on customer behavioral intention to use recommender systems,” *Inf. Syst. E-Bus. Manag.*, vol. 13, no. 4, pp. 769–799, 2015.
[6] D. Das, L. Sahoo, and S. Datta, “A survey on recommendation system,” *Int. J. Comput. Appl.*, vol. 160, no. 7, 2017.
[7] P. B. Thorat, R. M. Goudar, and S. Barve, “Survey on collaborative filtering, content-based filtering and hybrid recommendation system,” *Int. J. Comput. Appl.*, vol. 110, no. 4, pp. 31–36, 2015.
[8] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egypt. Inform. J.*, vol. 16, no. 3, pp. 261–273, Nov. 2015, doi: 10.1016/j.eij.2015.06.005.
[9] E. Rahm and H. H. Do, “Data cleaning: Problems and current approaches,” *IEEE Data Eng Bull*, vol. 23, no. 4, pp. 3–13, 2000.
[10] J. Horrigan, *Wireless internet use*. Pew Internet & American Life Project Washington, DC, 2009.