COMP4801 Final Year Project
AI Student Advisor
Supervised by Dr. T. W. Chim
Final Report
By: Chau Chun Wing (3035178765)
Other group members:
Chan Lap Kiu (3035269784)
Kwok cheuk Lum (3035269992)
Woo Chong Yiu (3035271012)
Date of submission: 13/4/2019
Abstract

Currently, many university students find it hard to choose courses [2]. While the students can seek advice from former students or even utilize several online course rating platforms, such as Triklo [1] these platforms often provide general raw data of how other students perform in a course and fail to make user-specific advice for individuals. Apart from that, existing platforms provide only functionality for users to comment on different courses, allowing each other viewers to view their comments and make their judgement on the course. These comments are merely the commenter’s thought on course, but not personal advice of the viewers.

Having the thought of such, a mobile application based on the principle of Recommender System (RS) is developed to solve the issue. Instead of providing non-user specific raw data and comments, this project focus on each users’ capabilities and characteristics to help them to choose courses. The entire system consists two major functionality, with the first one being a predictive machine learning model (K-nearest neighbors with WALS algorithm) utilizing users past course grade results to give future course grade predictions. The use of machine learning utilizes course grade result from many other students and provide user-specific course predictions, allowing them to choose the suitable courses to take in their study plan. Moreover, the academic advising community provides an online platform that allows students to raise and answer questions related to course selection and academic advice.

The deliverables for this project include an iPhone Operating System(iOS) application written in Objective C, an Android application written in Java, a web server developed with Node.js and a MongoDB database deployed respectively on Heroku and MongoDB Atlas, and a slightly modified machine learning model originated from Google Cloud[3].
Acknowledgement

We would like to express our deepest appreciation for those individuals and organizations, who have kindly offer their help and support to this project.

We are highly indebted to our supervisor Dr. T. W. Chim for his opinions and guidance throughout the project. He has always have meetings with us and has given a lot of concrete and innovative feedbacks. He also calls for people to finish our survey in collecting data and requests the department for data. We could not have achieved this much without his support. We would also like to thank Dr. S. M. Yiu for giving us valuable opinion during the intermediate presentation.

We would also express our gratitude to those who spare their time to complete the online survey and serve as data in the machine learning model.

We would also like to express our gratitude towards the Department of Computer Science, Faculty of Engineering, The University of Hong Kong for giving us the valuable opportunity for doing this project and providing us valuable data for model training, and Smitty Lam, the staff in the the Department of Computer Science, Faculty of Engineering, The University of Hong Kong for manually sorting out data for us.
# Table of Contents

Abstract  2
Acknowledgement  3
Table of Contents  4
List of Figures  6
List of Tables  7
Abbreviations  8

1. Introduction  9
   1.1 Background  9
   1.2 Previous works  10
   1.3 Objectives  10
   1.4 Scope  11
   1.5 Deliverables  11
   1.6 Outline of report  11

2. Methodology  12
   2.1 Development Approach  12
   2.2 System Architecture  13
   2.3 Presentation tier  13
   2.4 Logic Tier  14
      2.4.1 Server  14
      2.4.2 Prediction model  14
   2.5 Data Tier  14
   2.6 Functions Design  15
      2.6.1 Grade Prediction  15
      2.6.2 View posts in order and filter (Recent, Hot, Bookmarked and My Post)  15
      2.6.3 Search post  17
      2.6.4 Create post  17
      2.6.5 Bookmark post  17
      2.6.6 Add comment  18
      2.6.7 Authentication  19
   2.7 Model Training and Data Collection  21
      2.7.1 Collaborative Filtering  21
      2.7.2 Matrix factorization and WALS algorithm  22
      2.7.3 Data Collection  23
      2.7.4 Model Training and Cross Validation  23
### 2.7.5 Cold Start Problem and the solution

2.8 Project schedule

3 User Interface

3.1 Home View of forum and the Navigation drawer

3.4 Create post

3.5 Add comment/ view post and comment

3.6 Edit Post

3.7 Login/ Register

4 Results and Discussions

5 Encountered Challenges

5.1 Model Accuracy

5.2 Cold Start Problem

6 Conclusion

References
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An agile software development cycle diagram. In each development process, the circle is kept going though in every phases [6].</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Algorithm of calculating hot posts from Reddit [12].</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Sequence diagram of registration process</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>The sparse matrix that models users’ grade</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>The home page of AI Student Advisor</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Navigation drawer of AI Student Advisor</td>
<td>27</td>
</tr>
<tr>
<td>7</td>
<td>User interface of creating a post</td>
<td>29</td>
</tr>
<tr>
<td>8</td>
<td>User interface of adding a tag</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>A tag “COMP9999” being added</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>User interface of viewing post and comment</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>User interface of creating a comment</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>A comment being added</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>User interface of editing a post</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>User interface of viewing post and comment</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>User interface of creating a comment</td>
<td>32</td>
</tr>
<tr>
<td>16</td>
<td>A comment being added</td>
<td>32</td>
</tr>
</tbody>
</table>
List of Tables

Table 1. Schedule for this project. 26

Table 2: Table showing training model RMSE and MAE after filtering out 10 or below courses 33

Table 3: Table showing training model RMSE and MAE using OBL 33

Table 4: Table showing training model RMSE and MAE with no modification 33
### Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CF</td>
<td>Collaborative Filtering</td>
</tr>
<tr>
<td>CGPA</td>
<td>Cumulative Grade Point Average</td>
</tr>
<tr>
<td>CS</td>
<td>Computer Science</td>
</tr>
<tr>
<td>GPA</td>
<td>Grade Point Average</td>
</tr>
<tr>
<td>HKU</td>
<td>The University of Hong Kong</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>iOS</td>
<td>iPhone Operating System</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>OOP</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PaaS</td>
<td>Platform as a Service</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>RS</td>
<td>Recommendation System</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign On</td>
</tr>
<tr>
<td>NoSQL</td>
<td>Non Structured Query Language</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UX</td>
<td>User Experience</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Background

In the past, people often frequently had difficulty in choosing courses because information about courses was not readily open and available. When students are planning their study path and choosing courses, they often seek advice from seniors to choose what courses to take. Due to the prevalence of the internet, information about general course grades is easily accessible through online platforms such as Triklo [1]. These online platforms are categorized as online course rating platforms, allowing students to view general objective course grades by providing their own results. Students can also make comments on course workloads, teaching quality, and other details on individual courses, allowing other viewers to decide whether the course is good to take.

While providing an user-evaluated courses an objective general course result might be crucial to viewers to provide an first glance of the course, it is not sufficient as they are not user-specific with no consideration of one’s strength and weaknesses. On the other hand, while the platforms attempt to incorporate user subjective opinion by allowing users’ to comment on the courses might also be useful for course selection, they are again not advice to oneself but advice to course. These application go no further but only provide a general picture to user what a course is like, but not personalized advices to users.

Only providing general course grades may not be desirable for users with different capabilities and skills. For an algorithm course, user with great mathematical course might excel. For a law-related course, user with stronger language skills might prevail. The general course grades simply provides an average of how student perform in a course, but not considering each viewers strength and weakness. While these platforms try to provide viewers information to choose a course, the gap between general course grades and user strength fails them to achieve such goal.

The gap can be solved by the principle of Collaborative Filtering (CF), a technique commonly used by RS to generate user-specific content by collecting preferences or taste information from many users, which is often done with machine learning(ML).
In this project, it is used to generate user-specific course grade prediction by collecting or past course grade from many other users.

1.2 Previous works

There are existing works with similar ideas of producing an online academic advisor, but the technologies used and the main areas of concern are different. These works include UM-IBM SAPPHIRE PROJECT at the University of Michigan and A Proposed Model for a Web-Based Academic Advising System. For UM-IBM SAPPHIRE PROJECT, it aims at using AI technology to process natural language and change natural text into SQL [3] and apply it in an academic advisor so that it can reply the users’ natural language queries. Although this project also integrates the technologies of AI, the focus of this project is to provide a more convenient way for students to get advices but not giving them better advices. It also only improves the accessibility by resolving the users’ natural language queries but not through other means. The Proposed Model for a Web-Based Academic Advising System focus on reviewing what model should be used for an e-Academic Advising System in the modern day so that staff and advisors can follow-up the suggestions given and feedbacks from the students [4]. That project does not investigate on using any AI technologies to give better advices but only give suggestions on how web-based technologies can facilitate human advisors to follow-up on students’ feedbacks. Thus, it does not solve the problem of the suggestions not being accurate enough.

1.3 Objectives

The first objective is to use a machine learning model with the principal of a recommendation system in a form of mobile application that help students find courses they will perform well.

The second objective is to create an online community that allows students to discuss the course selection advice.

The third objective is to introduce intriguing social platform, gamified elements and attractive user interface (UI) to attract universities students to use it.
1.4 Scope

The mobile application is only developed to support iOS and Android platforms but not other platforms like Blackberry and Windows because these two platforms have already taken over 96% of the mobile OS market share [5]. Thus only developing on these two platforms is enough to cover most of the targeted users and meet the industrial standard. Owing to the difficulty in collecting past students’ grades from other departments and faculties, the mobile application only considers the courses that are held by the Computer Science Department in HKU during the current development stage.

1.5 Deliverables

A mobile application with both Android and iOS versions are the final products of this project. Feature of this application includes course grade prediction, course suggestions and an online community for course selection advice. The course grade prediction feature allows the user to predict his course grade for a particular course. The mobile application predicts the grade by using the data given by the user including his past course grades. A user can also seek for advices and ask questions in the online community if he feels more comfortable in asking his peers or advisors.

1.6 Outline of report

The remainder of this report proceeds as follow. The whole project’s development process will be discussed first. The methods on how to build a mobile application, the functionality of the mobile application and how to train the model will be discussed in chapter 2. Data collection methods and project schedule are also discussed. In chapter 3, the current user interface of the project are discussed. The result of evaluation on the AI model will be discussed in chapter 5. Then the difficulty we encountered during the development will be offered and discussed in chapter 5. Finally, it ended with a conclusion in chapter 6.
2. Methodology

2.1 Development Approach

![Agile Software Development Cycle Diagram](image)

*Figure 1. An agile software development cycle diagram. In each development process, the circle is kept going through in every phases [6].*

In this project, agile development process is used (see Figure 1) where the above iterative cycles have been repeated for times. It works by breaking the project into smaller functionalities to fit in each iterative phases, for example authentication, forum features and etc. In each iterative phase, the above iterative cycle is carried out from meet to evaluate. Upon the completion of each iterative phases, the functionality is evaluated and the system increments.

The iterative and incremental approach has several benefits over the traditional model such as the Waterfall model. The major benefit of the agile development process is that risk can be mitigated at early iterative cycles as early feedback can be obtained from stakeholders. Stakeholder’s requirements can be fulfilled by having a better understanding with their feedback, making early modification to the system. Waterfall model often fails to meet stakeholder requirements and often results in project delay due to its only one iteration. With only one iteration, risk cannot be discovered early and hence cannot be mitigated early, leading to project delay. Stakeholder requirements are also often poorly met due to failure to get early feedbacks.
2.2 System Architecture

Three-tier architecture is used in the project. Three-tier architecture is a software architecture pattern that divides the system into three tiers, usually the presentation layer, logic layer and data layer[3]. In such model, the logic layer and data layer are separated, which differentiate from the classical client-server architecture. Such approach reduces coupling and dependencies between layers, allowing development of layers to be carried out simultaneously, speeding up the development process. Apart from that, code can be easily maintained and are more scalable due to less coupling. The presentation tier consists of the well-defined user interface and some logic to present useful information to users. The logic tier is for communication between the presentation tier and data tier on data and the machine learning process of past course grades. The data tier is for data storage and retrieval.

2.3 Presentation tier

An iOS and Android application written in Objective-C and Java are developed in the system. According to Statista [4], Global market share held by the smartphone operating systems of iOS in 2018 is 11.9% and that of Android is 88%, making up a total of 99.9% of smartphone user. Both platform are chosen so as to accommodate most students in the university. Regarding programming language choice, Objective-C is chosen for iOS and Java is chosen for Android. While React native can be used to generate application in two platform with single implementation, it is rather new and not as stable as the traditional Objective-C and Java language. Also, iOS and Android users have different user needs and hence writing in two languages help to satisfy different needs of the two group of users. Apart from that, these two languages possess a larger community base, allowing problems to be sorted out quickly and easily. Some existing third party libraries and frameworks will be used to increase our development pace in frontend development. For example, AFNetworking and OkHttp are used on the iOS and Android platforms respectively to make HTTP requests and also handle the network errors effortlessly. DragRefreshAndLoadMoreTable and Ultra Pull To Refresh with Load More are used on the iOS and Android platforms respectively to implement the features of “pull to refresh” and “pull up to load more”.
2.4 Logic Tier

2.4.1 Server

Node.js is used for server development. It serves as a REST API server, allowing the presentation tier to access and manipulate representation of Web resources in the form of JSON, leaving the presentation of data to other tier to reduce coupling. Other popular server side languages include PHP and Django. Node.js was chosen because of its async functions and its non-blocking operation when comparing with the blocking operations in PHP and Django, which help in improving the server performance.

The server is deployed in Heroku, a cloud computing platform. A cloud computing platform is used as it can be managed by multiple administrators at different places easily at the same time. While other popular Cloud service such as Google Cloud and Azure exists, it is chosen due to limited budget of the project.

2.4.2 Prediction model

The model was trained on Google Cloud Virtual Machine Instance using the framework provided by Google Cloud with slight modification [5]. The reason of using such framework is that it enables faster development. It is written in Python with TensorFlow, which is a very popular choice for machine learning and also own a large community base.

2.5 Data Tier

MongoDB, a non-relational database is used. Comparing with traditional relational database such as MySQL, SQLite, it offers greater flexibility by storing data in documents with dynamic schema. Apart from that, MongoDB works well with the asynchronous function with Node JS.

The database server is deployed on MongoDB Atlas as it is the official cloud platform for MongoDB, meaning it is best supportive to the database. It is chosen also due to the fact of limited budget of the project.
2.6 Functions Design

The following sections describe the user-server interaction of the system. The mobile application consists of seven major functions: Grade Prediction, View posts in order, Search Posts, Create post, Bookmark post, Add comment, Authentication.

2.6.1 Grade Prediction

Users can get a course prediction from the mobile application after inputting his or her course.

The client application will send a request with the data to the server. The server will first run a k-nearest neighbor to map user to the nearest similar user in the model to address the cold start problem. Details of the Cold Start Problem will be demonstrated in 2.7. Then the server will use the most similar user to retrieve the predicted grade server from the model (using framework in Google Cloud) trained in Google Cloud. Finally, the server will send the list of 27 courses course result back to the mobile application and The mobile application will show the grade prediction result to the user with its front-end logic.

A solution provided by Google Cloud Platform(GCP) Building a Recommendation System in TensorFlow has been used as the machine learning program in this project to predict course grade due to the lack of experience in and AI field and limited time for the project. Data is uploaded to the Google Cloud Virtual Machine Instance and the model is trained. The model is then saved and tuned, and finally in production after incorporating it into the backend.

The detailed data collection, model training and theoretical background method will be discussed in section X.X.

2.6.2 View posts in order and filter (Recent, Hot, Bookmarked and My Post)

Users can view posts by specifying one of the four types of view orders and filters, Recent, Hot, Bookmarked and My Post.
The client application sends a request to the server with specific view orders. Then, the server then sends a Database request to the Database server, and retrieve the first 10 posts records with the specify view order. When the user scrolls to the bottom of the mobile application, the client will send another request for the following ten posts with specify view order. The requests can be repeated until all matching posts are return from the server.

\[
\text{Given the time the entry was posted } A \text{ and the time of } 7:46:43 \text{ a.m. } \\
\text{December 8, 2005 } B, \text{ we have } t, \text{ as their difference in seconds}
\]
\[
t = A - B
\]
and \( x \) as the difference between the number of up votes \( U \) and the number of down votes \( D \)
\[
x = U - D
\]
where \( y \in \{ -1, 0, 1 \} \)
\[
y = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{if } x = 0 \\
-1 & \text{if } x < 0
\end{cases}
\]
and \( z \) as the maximal value, of the absolute value of \( x \) and 1
\[
z = \begin{cases} 
|x| & \text{if } |x| \geq 1 \\
1 & \text{if } |x| < 1
\end{cases}
\]
we have the rating as a function \( f(t, y, z) \)
\[
f(t, y, z) = \log_{10} z + \frac{yt}{45000}
\]

*Figure 2. Algorithm of calculating hot posts from Reddit [12].*

In particular, the view hot posts are ranked by the formula (see Figure 2), references from the algorithm from Reddit, a popular open source forum in the world. The variable \( x \) is modified from the number of upvotes and downvotes to the number of comments in this mobile application because there are no upvotes and downvotes in this mobile application. In this algorithm, submission time plays an important role where newer stories will be ranked higher than older ones. The newest posts will almost always be ranked at the top. Another feature of the algorithm is that the deteriorating effect of the number of comment as logarithm of comment is used.
2.6.3 Search post

Users can search a post by providing keywords.

User first enter the keyword in the search bar and clicks “search”. The client will send a search request to the server with specified keyword and session id. The server will then send a Database retrieval request to the Database server. Upon receiving records from the Database server, the Database server will send the first 5 matching post back to the mobile application. When the user scrolls to the bottom of the mobile application, the client will send another request for the following ten posts with specify view order. The requests can be repeated until all matching posts are return from the server.

2.6.4 Create post

Users can create a post related to course selection.

User first clicks the create post button first, and followed by entering the title, content, and tags. The client will send a create post request with title, content, tags and session id to the server. The server will then send a write request to the Database server and return a success message if it successfully adds the document to the database. Upon receiving the success message, the client application will return to the home page, showing the post just added by the user.

2.6.5 Bookmark post

Users can create a bookmark to the post for following up of the post.

User first clicks the bookmark button of any post in the client-side application. The application will send the bookmark post request with corresponding post id and session id. The server will then send a request to the Database and add the user id to the post document in the database. Upon successful creation of record, the server will then send a success message to the client mobile application. If the client-side receive the successful message, the bookmark button is blackened.
2.6.6 Add comment

Users can add comments to discuss a post related to course selection.

Userpost when they enter a post in the client-side application. The user only needs to fill in the content of the reply, then the client-side application will send a reply post request to the backend server with user id and post id. After the backend server receives the reply post request, it will insert the document into the comment database, and return a successful message if record inserted successfully. If the client-side receive the successful message, the client-side application will append the comments on the page directly, while not request the backend server to send all the comments in the post again as it will waste some performance of the backend server. Otherwise, the client-side will pop up an error message according to the error message received from the backend server.
2.6.7 Authentication

The authentication function allows users to register or log in the application. The user cannot use other functions before logging in.

![Sequence diagram of registration process.](image)

For the registration process, the user needs to fill in his username, password, and other personal information in the mobile application. The entered username and password are first validated by the client-side application, so that they are not empty and the password is longer than six characters. Then the client sends the registration request to the backend server. The server first checks whether there is a duplicate username in the database, then hash and encrypt the password and insert the document into the database. The server then sends an email containing a one time token to the user’s email to verify his or her identity. In order to prevent hackers from guessing the one time token, this unauthenticated account and one time token are only stored in the database for 15 minutes. If the user does not enter and submit the correct token within the time period, he or she needs to start the registration process again. If the one time token verification is successful, the user is automatically logged in and can start using
the application directly. Passwords are hashed before saving into the database. By hashing a password, a hacker cannot recover the original password even if he or she has access to the value stored in the database.

Bcrypt is used to hash and encrypt the passwords rather than other alternatives in the market as it was specifically designed for passwords and thus have better performance. Furthermore, a 128-bit salt [13] is included and the iteration count can be modified in the algorithm of bcrypt so it is immune to rainbow attacks and brute force attacks.

To login, the user needs to enter the username and password that he or she used to register previously. The application first validates the input and then send the login request to the server. Then backend checks if both the username and password are same as the database by hashing and encrypting the received password using the bcrypt algorithm again. Finally, the server will return a successful message if both username and password are same as the database. If the client-side receive the successful message, the client-side application will login and redirect user to the home page of the application which shows the latest posts. Otherwise, the client-side will pop up an error message according to the error message received from the backend server.
2.7 Model Training and Data Collection

The following section demonstrates the machine learning model and means for data collection used in machine learning.

2.7.1 Collaborative Filtering

The model takes in other user’s past course grade result and predict a new user’s course grade with the new user’s already achieved grade. The model is insighted from the powerful method Collaborative Filtering to generate approach in Recommender System. It is noticed that the aim of the project is to predict GPA, which is very similar in the Collaborative Filtering approach. The approach originally returns item(course) recommendation base on the highest predicted rating to the user. In this project, the predicted course grade will be in the model will be returned to the user.

In Collaborative Filtering, the basic assumption made is “if a person A has the same opinion as a person B on an issue, A is more likely to have B’s opinion on a different issue.” In this project, we assume that “if a person A has the same opinion as a person B on a course grade, A is more likely to have B’s course grade on a different course.”
2.7.2 Matrix factorization and WALS algorithm

The collaborative filtering problem can be solved using matrix factorization algorithm. The problem itself is model into a large Sparse matrix $M(m, n)$, where there are $m$ users, and a total of $n$ courses. As each user has individual set of pass course grades, the particular $n^{th}$ course grade of the $m^{th}$ user is stored in entry $M(m,n)$. If a user has never studied a course before, the entry is zero. The vast majority of the Sparse matrix entry are often 80% zeroes [4].

The job here is to utilize the matrix factorization method to predict those entries that are zeroes. The matrix factorization method assumes that there is a set of attributes common to all courses, with courses having different magnitude of the attributes. For example, in the Computer Science Curriculum, courses are categorized with different Outcome-Based Learning (OBL), where different courses have different weighting of different OBLs. By the assumption, the user’s grade can be predicted by considering a users strength from their course data to express their OBL outcomes. The
Attributes(OBL) are called latent factors. As there are ten OBLs, there are a total of ten latent factors as assumption.

The large Sparse matrix $M(m,n)$ is then approximately decomposed into two matrices with dimension $(m, 10) \times (10, n)$. The inner product of the two smaller matrices approximates the original matrix, predicting the original zero entries in the original matrix. The algorithm used in this is called the WALS method of matrix factorization. While many other methods such as SVD and ALS can be used, WALS excels in optimize so that the two matrices update is very efficient, reducing the training time. It is noted that framework from Google Cloud Solution *Building a Recommendation System in TensorFlow: Overview* is used rather manual implementation of the algorithm due to limited time [5].

### 2.7.3 Data Collection

Data of Computer Science Student undergraduate past course grades are collected to train the model. The quota for undergraduate place in the Department of Computer Science of HKU is 120 [12]. With consideration of the change in curriculum, 10 years is chosen at the time of data, yielding a sample size of 1200 students. With mathematical calculation 291 students are needed for 95% confidence interval.

Two methods will be used to collect the initial data. First, questionnaires will be distributed online to student of their individual course grades. First, grades, including the GPA and individual course grades, of past graduates are collected from the Department of Computer Science of HKU. Second, questionnaires will be distributed online to student for their individual course grades.

### 2.7.4 Model Training and Cross Validation

The Data collected with the above means will be used for the model training mentioned above with Google Cloud VM Instance. In the implementation, the whole data set is divided into train set and test set with a ratio of 90%/10%. The train set is used to train the model while the test set is used to calculate the accuracy of the prediction. The metric Root Mean Square Error (RMSE) and Mean Absolute Error(MAE) is used to validate the accuracy of the trained model. In particular, the MAE can be used calculate the interval of the prediction. If the predicted grade of a
course is 3.3 and the model has a MAE of 0.5, the possible range of the predicted grade is (2.8, 3.8). A typical acceptable value of RMSE is around 0.6 and MAE of around 0.5. After the training is done and accuracy is calculated. The model is tuned for its hyperparameter so as to reduce the RMSE and MAE.

### 2.7.5 Cold Start Problem and the solution

After the tuning of model, it will then be deployed in the original server for data prediction. However, the limitation of the model is that it can only predict existing users that are used in the training model. If a new user requests for prediction from the server, the whole model has to be retrained and tuned, which is known as the Cold Start problem. It is not possible to train and tune the model in real time as it takes at least 10 minutes to do so, which is infeasible for a long waiting time. To address the cold start problem, the new user is the first mapped to the most similar course grades user used in the training model with K-nearest neighbor where k=1. It will then feed old user’s identity to the trained model and make grade prediction.

### 2.8 Project schedule

<table>
<thead>
<tr>
<th>Task</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface of register</td>
<td>30 October 2018</td>
</tr>
<tr>
<td>Interface of Log in and out</td>
<td></td>
</tr>
<tr>
<td>Deploy server and database at cloud</td>
<td></td>
</tr>
<tr>
<td>Server logic of register</td>
<td></td>
</tr>
<tr>
<td>Server logic of Log in and out</td>
<td></td>
</tr>
<tr>
<td>Interface for forum</td>
<td>30 December 2018</td>
</tr>
<tr>
<td>Logic of all forum features</td>
<td></td>
</tr>
<tr>
<td>Research on Machine learning</td>
<td>30 January 2019</td>
</tr>
<tr>
<td>Data collection and integration</td>
<td></td>
</tr>
<tr>
<td>Interface of Online survey</td>
<td></td>
</tr>
<tr>
<td>Server logic of Online survey</td>
<td></td>
</tr>
<tr>
<td>Deploy machine learning</td>
<td>30 February 2019</td>
</tr>
<tr>
<td>Train and tune the model</td>
<td></td>
</tr>
<tr>
<td>Interface for course grade prediction</td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Date</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Testing and Debugging</td>
<td>10 April 2019</td>
</tr>
<tr>
<td>Deliverable of code and Final Report</td>
<td>14 April 2019</td>
</tr>
<tr>
<td>Final Presentation</td>
<td>22 April 2019</td>
</tr>
<tr>
<td>Exhibition</td>
<td>28 April 2019</td>
</tr>
</tbody>
</table>

*Table 1. Schedule for this project.*
3 User Interface

3.1 Home View of forum and the Navigation drawer

Figure 5: The home page of AI Student Advisor
Figure 6: Navigation drawer of AI Student Advisor.

Forum post is shown in Figure 5, which is the homepage of the mobile application. The homepage can also be accessed by the “Forum” button navigation drawer in Figure 8.

For each posts in the homepage, the post title, content, post tags, number of comments ( ), and a bookmark button ( ). The bookmark button is used to bookmark or unbookmark a post. When the post is bookmarked, it is in the blackend form ( ). The “+” button on the top right corner of the home page is used when user would like to raise a new question on course selection. User can also click in a particular post for viewing the full content of a post or leave a comment under the post.

As the user scroll to the very bottom of the application, more posts are received from the server and is displayed in the home page.
The page “My Posts”, “Bookmarked” and “Hot” are very similar to the Homepage and the only difference is the sort order and criteria of displaying the post. “My Posts” tab display all the post created by the user, “Bookmarked” displays all the post bookmarked by the user, and “Hot” tab displays all the post in ranked hotness described in the Methodology section.

All posts shown in Figure 5 is stored at local base, and hence internet connection is required. The homepage can only be accessible with authentication.

For the navigation drawer, it is used to navigate to other views of the mobile application. It is divided into three main parts. The topmost part is the brief profile of the user with his nick name (and CGPA) and photo. The top left corner (←) of the navigation drawer is used to close the navigation drawer. The middle part consists of the three criteria to view posts and user to click on one of the tabs to view posts according to the criteria. The bottom part consists of the functions of the application, the home page showing all the posts of course selection, the grade prediction tab allows user to predict grade by entering his or her own, the setting tab allows user to modify the general setting of the mobile application and the sign out tab allow user to sign out of the system.
3.4 Create post

When a users click on the add post button (+) at the top right corner on any page of the online community (see Figure 5), the app will create a new screen of creating a post (see Figure 9). User has to fill in the post title in the top most text box, the content of the post in the second text box, any tags that the user would like to tag, and then the “Post” button on the top right hand corner. Upon succession of creating the post, the app will direct to the home page (see Figure 5).

When user clicks the “Add” button in the screen, a dialog box is popped up (see Figure 10). User enters the tag they would like to be added into the text field of the dialog box and can complete adding a tag by clicking “Add” or return to the previous screen (see Figure 9) without creating a tag by clicking “Cancel” in the dialog box. Tags successfully created will be listed below “Tags” in the oval form (see Figure 9). Tags can be deleted by clicking on the displayed created tags. Multiple tags can be created by repeating the above procedures.
3.5 Add comment/view post and comment

Figure 10: User interface of viewing post and comment.

Figure 11: User interface of creating a comment.

Figure 12: A comment being added.

When user clicks on a post on the homepage (see Figure 5), the app redirects to the screen of viewing posts and comments, and adding comments (see Figure 12). This view shows the details of the post including the post author, title of the post, content of the post, the number of comments comment icon ( ), bookmark button ( ) and a text field. (see Figure 12). The bookmark button ( ) has the same functionality of the bookmark button ( ) in the homage (see Figure 5).

Under the post sits the comments (see Figure 5a). No comments are shown in this area if there are no comments( see Figure 12) Each comment shows its creator, content and the number of upflow(↑) and downflow(). Users can create comment at the edit text box located at the bottom of the view(see Figure 13). The keyboard pops up after tapping the edit text box and is pushed up by the keyboard (see Figure 13). Users can enter the the comment in the edit text box and submit the comment using the send button ( ).After submitting the comment, the virtual keyboard is hidden and the newly created comment is added at the bottom of the list (see Figure 14). The screen will automatically scroll to the bottom to display the newest comment if it was not.
If the user is the creator of the post, an edit button ( Nuevo ) is shown in the top right corner (see Figure 5a). Users can edit the post by pressing the edit button. An edit post screen will be launched. Details of the edit post screen will be discussed in the next section.

3.6 Edit Post

![Image of edit post interface]

**Figure 13: User interface of editing a post**

User can edit a post in the edit post screen (see Figure 12) navigated from the comments screen (see Figure 12). The screen is nearly the same as the create post screen (see Figure 9). The only difference is that it automatically fills in the original post title in the smaller text box, the original post content in the bigger text box, and the original tag(s) at the bottom.
3.7 Login/ Register

The app shows the login screen (see Figure 16) when users launch the app for the first time or after logging out. Users can login by pressing “Login” button after entering their correct username and password.

If a user does not have an account, he or she can click “Register” in blue at the bottom of the screen (see Figure 16). Then the registration screen is shown (See Figure 17). After filling in his or her email and password, he or she can click “Continue” in blue at the bottom of the screen to navigate to the next step (see Figure 18). The view then prompt the user to enter the one-time token which is sent through email. The user has to enter the token into the text box and click “Continue” in blue under the text box. They can click “Resend token” in blue at the bottom of the screen if the token is lost.

If the email is validated, the registration process is completed and the application redirects the user to the home screen.

4 Results and Discussions

A total of 100 students past course grade is received from the Computer Science Department and a total of 28 students record from the online survey is collected,
adding up to a total of 128 students from the online survey.

Three different preprocesses are carried out before feeding the collected students’ pass grade into the training model. The first approach is to filter out students that has enroll in less than 10 computer science course. The reason for this is to filter out the very sparse value in order to achieve a smaller error. The second approach is to reduce the 27 computer science course item into the 10 OBLs provided by the Computer Science Department. This is again to reduce the sparsity of the rating matrix. The last approach is to carry out the model training without any modification of the data set. The result of the three approaches are listed in the table below. Each training is carried out for 10 times.

<table>
<thead>
<tr>
<th>Filter out 10 or below courses</th>
<th>Train RMSE</th>
<th>Testing set RMSE</th>
<th>Train MAE</th>
<th>Test MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.502</td>
<td>0.687</td>
<td>0.398</td>
<td>0.547</td>
</tr>
<tr>
<td>range</td>
<td>0.49 - 0.51</td>
<td>0.67 - 0.78</td>
<td>0.39 – 0.4</td>
<td>0.49-0.61</td>
</tr>
</tbody>
</table>

*Table 2: Table showing training model RMSE and MAE after filtering out 10 or below courses*

<table>
<thead>
<tr>
<th>Using OBL</th>
<th>Train RMSE</th>
<th>Testing set RMSE</th>
<th>Train MAE</th>
<th>Test MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.513</td>
<td>0.67</td>
<td>0.42</td>
<td>0.585</td>
</tr>
<tr>
<td>range</td>
<td>0.47 – 0.52</td>
<td>0.67 - 0.89</td>
<td>0.4 – 0.43</td>
<td>0.54 - 0.64</td>
</tr>
</tbody>
</table>

*Table 3: Table showing training model RMSE and MAE using OBL*

<table>
<thead>
<tr>
<th></th>
<th>Train RMSE</th>
<th>Testing set RMSE</th>
<th>Train MAE</th>
<th>Test MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.516</td>
<td>0.798</td>
<td>0.408</td>
<td>0.597</td>
</tr>
<tr>
<td>range</td>
<td>0.51 – 0.52</td>
<td>0.68 – 1.05</td>
<td>0.4 – 0.41</td>
<td>0.56-0.69</td>
</tr>
</tbody>
</table>

*Table 4: Table showing training model RMSE and MAE with no modification*

The prediction results are broken down by the three approaches. Unexpectedly, it is noticed that the reduction of sparsity in OBL (see table 3) did not help much of the error comparing to no modification of data (see table 4), the improvement is only
0.012 in the test MAE. The accuracy did improve a lot by filtering out students that has studied 10 or less courses by 0.05.

The whole model is originally trained with 65 students and the average test MAE is around 0.602. The MAE has not improve much even if the past course grade size is around double. This shows that the MAE might be stuck in a local minimum and hence the MAE is not improved much.
5. Encountered Challenges

In this section, the encountered challenges and the corresponding solutions or proposed solution are described.

5.1 Model Accuracy

Problem:

The insufficient collection of data might affect the accuracy of the model. In the Google Cloud Solution, the dataset used that demonstrated the WALS model is the movieLens test set has around 100000 users with an test RMSE of 0.89 with range [0, 5] after tuning. However, in view of the collection of at most only 300 students past course result, the error might be very high.

Solution:

Data are preprocessed by reducing the sparsity of the matrix. This is done by filtering out all the student that take less than 10 courses or map courses to OBLs. Fortunately, the resulting error does not affect much from the result section.

5.2 Cold Start Problem

Problem:

The WALS collaborative filtering model is by its nature hard to predict new user’s grade or it need to be trained at real time to return the predicted grades. However, it takes at least 10 minutes, which is infeasible and non-user friendly for a mobile
application.

Current Solution:

K-nearest neighbor is used to evaluate the similarity between the new user and the old data. The new user is then map to the old user and predicted grade is return using the old user’s past grades. However, this solution is not perfect as the introduction of similarity further reduce the accuracy of the predictions.

Proposed Solution:

Add a cron job that train the model every night so that new user is incorporated in the WALS model. In the user interface of grade prediction, acknowledge the users to look at a more precise result in the next day.

6. Conclusion

Predicting grades of student with past students dataset is the key part in the project, helping students to make a more rational and suitable course selection. Instead of providing the general course data provided by the online platform, the course prediction feature and the forum centers the user around, better at helping them to make a good course selection.

The smartphone application has been fully implemented. It consists of 7 major features including Grade Prediction, View posts in order, Search Posts, Create post, Bookmark post, Add comment, Authentication. Although the amount of data collected might not be satisfactory, ultimate MAE of the model is only around 0.6, which is only two subgrades.
The initial objectives for this project have been achieved, but there are possibilities for future developments that can focus on other aspects. First, personality, the original thought can also be incorporated into the model as it might be a factor affecting course grade. Second, the scope of the project can be extended to HKU courses or even University course if enough records can be collected to benefit more university students to choose a course.
References


[2] Sing Tao Daily, “大學學生選科後中途變卦 僅三分一在六年內畢業 ;” 大學生選科後中途變卦 僅三分一在六年內畢業, Jan. 20, 2017. [Online]. Available: http://std.stheadline.com/daily/article/detail/1537743-%E6%B5%B7%E5%A4%96%E8%8F%AF%E4%BA%BA-%E5%A4%A7%E5%AD%B8%E7%94%9F%E9%81%B8%E7%A7%91%E5%BE%8C%E4%B8%89%E5%88%86%E4%B8%80%E5%9C%A8%E5%85%AD%E5%B9%B4%E5%85%A7%E7%95%A2%E6%A5%AD. [Accessed: Oct. 16, 2018]


