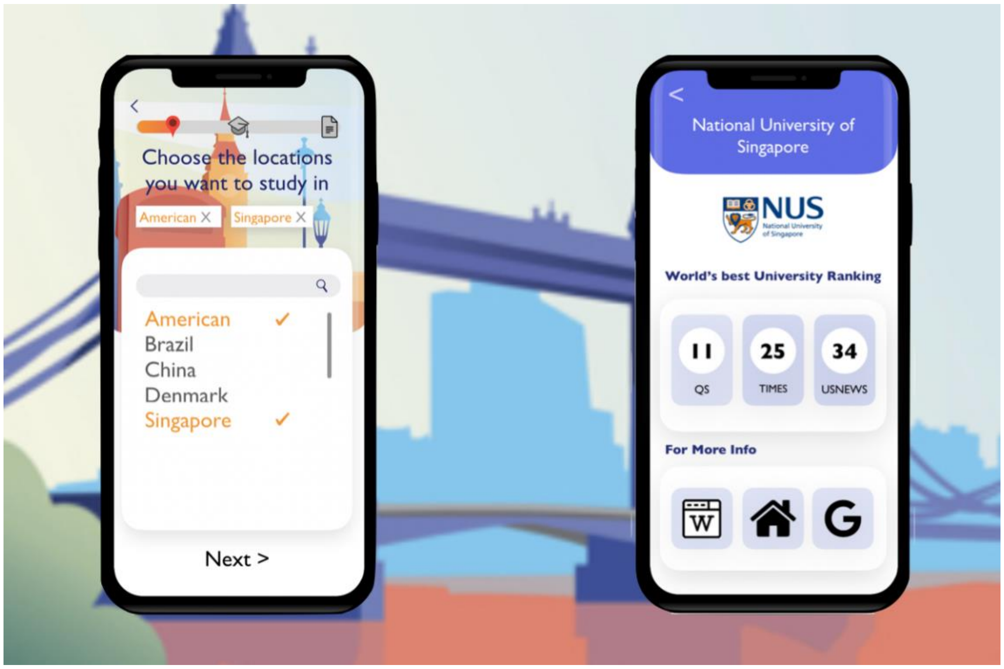

Choose the locations you want to study in
American
Singapore
American
Brazil
China
Denmark
Singapore
Next >
National University of Singapore
NUS
National University of Singapore
World's best University Ranking
11
QS
25
TIMES
34
USNEWS
For More Info

School Recommender Project Report

NUS Master of Technology
Intelligent Reasoning System (IRS)

Brought to you by Group 4

Cao Wen A0215516L
Lin Xi A0215403W
Liu Chenxi A0215461M
Zheng Kai A0215414R

20-Oct-2020

Table of Content

Table of Content 2

1 Executive Summary 4

2 Business Justification 5

3 Project Team 6

3.1 Project Objective 6

3.2 Team Members 6

4 Project Solution 7

4.1 Project Deliverables 7

4.1.1 Application Features 7

4.1.2 Application Business Flow 8

4.2 Knowledge Representation 9

4.2.1 Data Acquisition and Cleansing 9

4.2.2 Database Structure 10

5 Project Architecture & Implementation 11

5.1 Architecture Overview 11

5.2 Process Flow 11

5.3 System Modules 12

5.3.1 Decision Table 12

5.3.2 Decision Tree 13

5.3.3 Retrieval Model 14

5.3.4 User interface 17

5.3.5 Backend 18

6 Project Performance & Validation 18

6.1 Dataset and Evaluation Matrix 18

6.2 Evaluation 19

7 Challenge & Recommendation 21

7.1 Challenges 21

7.1.1 Data Acquisition via Web Crawling 21

7.1.2 Reaching out to the Public 21

7.2 Future Improvements 22

7.2.1 Better Data Acquisition 22

7.2.2 Better and More Accurate Decision Models 22

APPENDIX OF REPORT A 23
APPENDIX OF REPORT B 30
APPENDIX OF REPORT C 32
APPENDIX OF REPORT E 33
Individual Report: Cao Wen (A0215516L) 34
Personal Contribution 34
Learning outcome 34
Knowledge and Skill Application 34
Individual Report: Lin Xi (A0215403W) 36
Personal Contribution 36
Learning outcome 36
Knowledge and Skill Application 37
Individual Report: Liu Chenxi (A0215461M) 38
Personal Contribution 38
Learning outcome 38
Knowledge and Skill Application 38
Individual Report: ZHENG Kai (A0215414R) 40
Personal Contribution 40
Learning outcome 40
Knowledge and Skill Application 40

1 Executive Summary

In 2019, the number of overseas Chinese students reached 710,000. The United States is still having the largest number of Chinese students. The number of students studying in the UK has increased significantly, the number of students studying in Asia, Singapore, Japan, and South Korea has steadily increased. Countries studying abroad for Chinese students are becoming more and more diverse and the group of students studying abroad is growing.

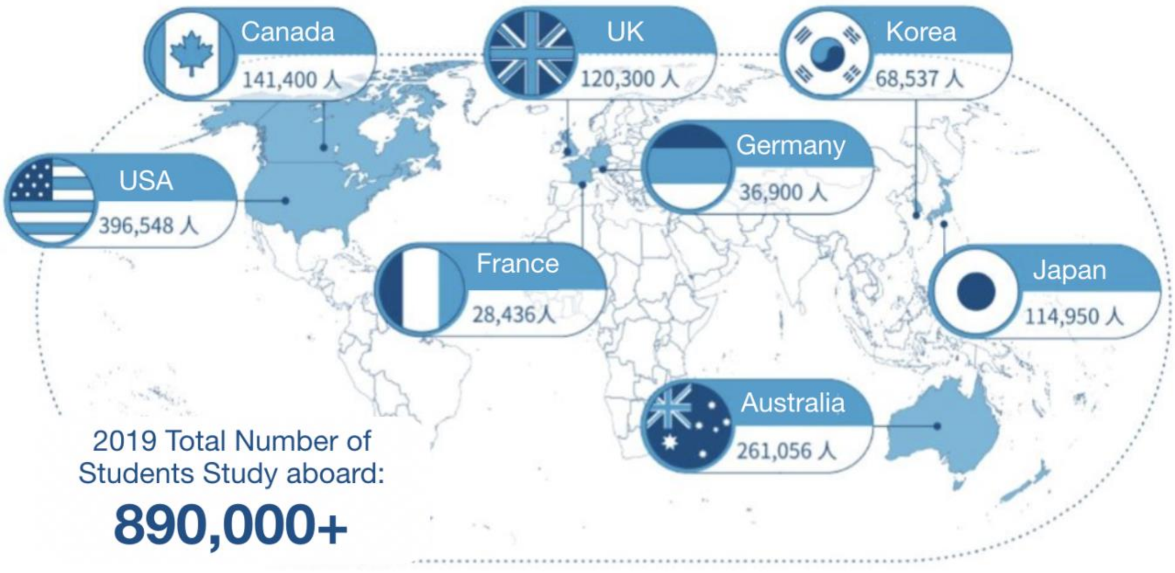


Figure 1-1: 2019 Chinese Students Study Aboard Statistic

Indeed, Most Chinese students decide to study abroad after obtaining their bachelor's degree in China. Students in China usually apply for overseas Mater or straight to Ph.D. programs to further develop their potential as intellectual leaders for a wide range of career paths based on their undergraduate school, GPA, TOEFL/IELTS/GRE, and other experience.

There are usually three ways for students to collect information and choose their target school. To begin with, the most common way is by themselves. This seems to be a fast and economic mode, however, due to lack of experience and limited energy for a single person, it’s difficult and time-consuming for them to find useful information from a large amount of college information. The Second way is seeking help from experienced relatives and friends. They can provide truthful and in-depth information, but too often their information only applies to their own background and maybe out of date. Last but not least, consulting a study abroad agency is the most efficient way to find the right school for you since they have been equipped with rich experience in a variety of cases. But the consulting fee is often too high to afford for most students.

Our project team is dedicated to assisting students to find the most suitable colleges and majors by taking advantage of the rich case experience of study abroad agencies. First We collected study abroad application data from study abroad agency and apply machine reasoning to prepare our knowledge base and evaluate students' backgrounds. Secondly, we have made use of the decision tree and retrieval model to match the student's background with that in our knowledge base to help students find out the most suitable school and major. Further to that, we have provided the QS, TIMES, USNEWS world university ranking data, and related information for the target school.

Our project team hopes that with our solution, the students will be able to find the school that suits their specific background most.

2 Business Justification

At the start of our team's journey, there were not many notable applications that were specifically catered to providing useful information and guidance on choosing a master school for an undergraduate. At this point in time, the threat of the COVID-19 has been under control in many countries, international communication is gradually recovering and the number of the student preparing to study aboard is increasing. Thus, we felt that there was a gap to be filled in the market and we set out in developing this system that would fill this need for choosing a suitable target school and overseas school related information to be accessed in a convenient and efficient method.

A study abroad agency can also integrate our systems with their own platform at the same time, provide a quick and preliminary assessment and recommendation for students, expand their business, and get more users. This knowledge-based system, serving as an experienced consultant, can provide professional recommendations and evaluations based on a large number of high-quality history application cases, which can save a lot of human resources cost in acquiring new users.

3 Project Team

3.1 Project Objective

The main objective of this group project is to develop a school recommender system, which is capable of recommending appropriate school and major for students based on high-quality study aboard application cases data, providing professional assessment for the student's background and easy access to target school-related information.

3.2 Team Members

Full Name	Work Items (Who Did What)
Cao Wen	• Project idea generation • Data Acquisition – crawl JiTuo study aboard application data, , collect school ranking info • Data Processing – process GPA/language test/target Major data • Decision Tree development • User Interface development – user info collection pages • Django & Database development – background assessment API, get recommendation school API, system integration • Debug & troubleshooting • Project management • Project report writing
Lin Xi	• Product prototype design – product solution, business flow design, UI design • System architecture design – system/data flow design, system modules design, database structure design • Data Processing – extract competition/research information from raw data, collect school ranking info • User Interface development – index, assessment report, school detail page development, all pages layout adjustment • Django & Database development – user database management, get school detail API • Project management • Debug & troubleshooting • Project report writing
Liu Chen Xi	• Project idea generation • System architecture design – system/data flow design, system modules design • Data Acquisition – crawl BUPT study aboard application data, , collect school ranking info • Data Processing – process undergraduate school/major, training data augmentation

	• Retrieval model development – model training, hyperparameter tuning, model evaluation • User Interface development – country select pages • Debug & troubleshooting • Project report writing
Zheng Kai	• Project idea generation • Data Processing – process target school data, application result data process • Decision Table – expert rule survey and design, module development • Decision Tree development • Debug & troubleshooting • Project report writing

4 Project Solution

4.1 Project Deliverables

4.1.1 Application Features

The school recommender system aims to make good use of the history study abroad application data to recommend schools for students that match their undergraduate background. Besides, the system is expected to have several features such as providing assessment for the user's background such as undergraduate school level, GPA, etc., comparing these features with history application data in our database, providing the position of the applicant's background among all the applicants in database, providing recommendations for school & major and the easy access to the school information. Below shows the diagram which summarizes the project deliverables, where the system consists of four main features.

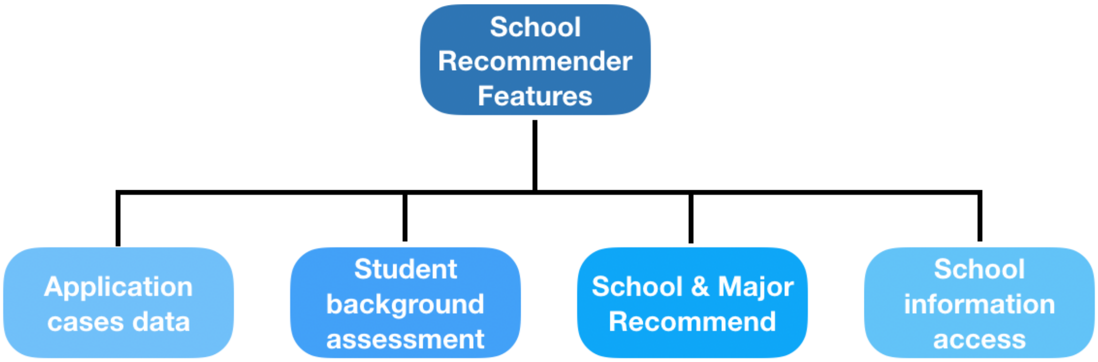

School Recommender Features
Application cases data
Student background assessment
School & Major Recommend
School information access

Figure 4-1: School Recommender Feature

Application case data: Collect comprehensive history application data for students from different undergraduate schools and major, contains more than 10 popular studies aboard destination.

Student background assessment: assess the student's competitiveness by expert rule in each dimension, such as undergraduate school/GPA/TOEFL/IELTS.

School & Major recommends: based on the student's background information and the application history records, find out the school & major suits him/her most.

School information access: provide quick and easy access for the school’s ranking info and some other related websites.

4.1.2 Application Business Flow

Figure 4-2 shows the business flow of our school recommender system.

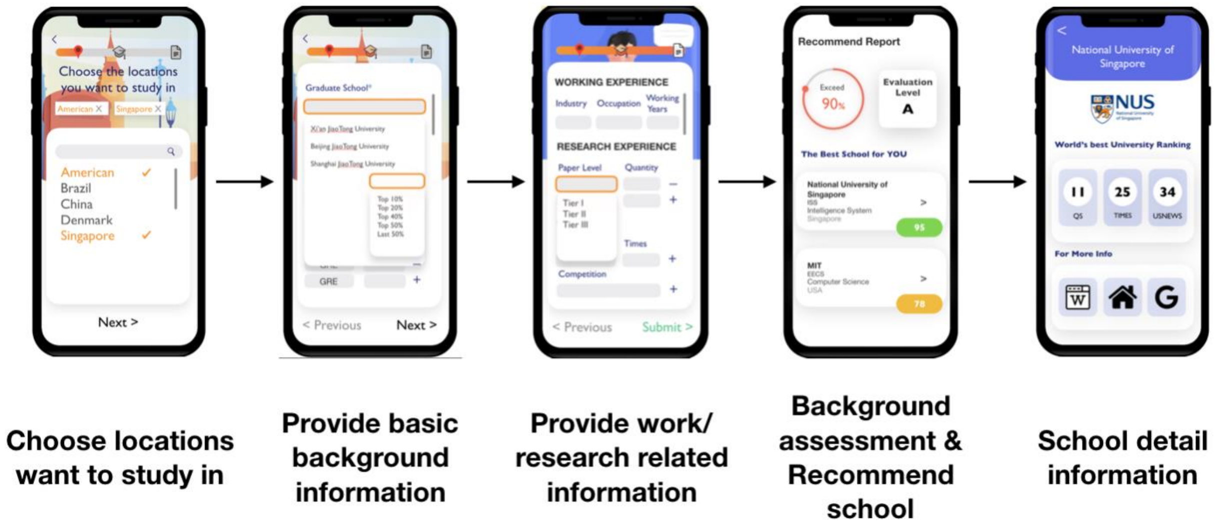

Choose the locations you want to study in
American
Brazil
China
Denmark
Singapore
Next >
Graduate School
GRE
< Previous
Next >
WORKING EXPERIENCE
RESEARCH EXPERIENCE
Submit >
Recommend Report
Exceed 90%
Evaluation Level A
The Best School for YOU
National University of Singapore
MIT
National University of Singapore
NUS
World's best University Ranking
11
25
34
For More Info
Choose locations want to study in
Provide basic background information
Provide work/ research related information
Background assessment & Recommend school
School detail information

Figure 4-2: Application Business Flow

Our system is basically a knowledge-driven system, as the machine reasoning and machine learning model will focus on delivering accurate and reliable results by utilizing the knowledge. When a user finishes providing their background information through our user interface, the frontend will send these data into Django and process these data via decision table into structure data and assess this information then return the assessment results. These data will be fed into a decision tree and retrieval model. A decision tree will generate the overall evaluation level for the user, and a retrieval model can find out the most suitable school & major pair for the specific user. Finally, the system uses API to retrieve the specific school information from the MYSQL database.

4.2 Knowledge Representation

4.2.1 Data Acquisition and Cleansing

4.2.1.1 Extraction Method

This system needs students application records to build a reliable predict model and overseas universities' introduction to provide some basic information about the recommended school. The table below shows the list of web pages where data are to be extracted.

Table 4-1: Data resource website and description

Website	Data to Crawl
http://bbs.gter.net/	Students’ application records, including their academic performance, working experience and their application results.
https://www.topuniversities.com/	Worldwide university rankings
https://en.wikipedia.org/wiki/	Basic information of these universities

Below shows the flow chart on how the extraction of data is done in our recommend system.

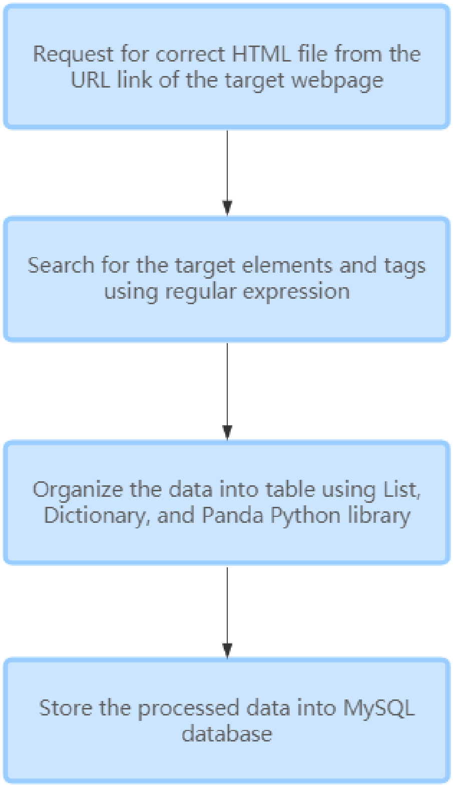

Request for correct HTML file from the URL link of the target webpage
Search for the target elements and tags using regular expression
Organize the data into table using List, Dictionary, and Panda Python library
Store the processed data into MySQL database

Figure 4-3: Data Extraction Flow

To extract resources and data from online web pages, our system will need to have a web scraping function to do the job. One of the best ways to define some patterns using regular expressions to search for information we're interested in. According to the target web pages’ HTML, corresponding patterns can be defined to extract needed texts. We use the “findall” method in the RE library to do the most extracting job, which makes use of elements and tags in the HTML file for the searching.

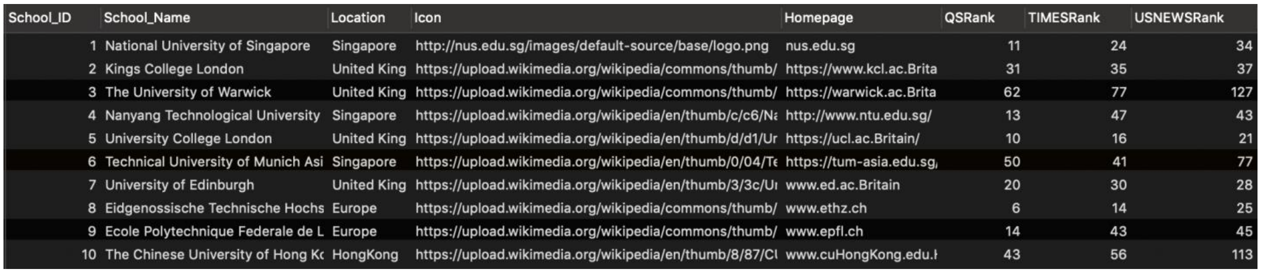
After extracting string data from an HTML file, the system will organize the data into tables using various data containers such as list, dictionary from standard Python Library, and DataFrame from Panda Python Library. The table will then be stored in MySQL Database, which is accessible in the later stage.

4.2.1.2 Cleaning Method

To determine inaccurate and incomplete data and then improve the quality of data we crawled from webpages, data cleaning is required to be executed in the pre-processing stage. For inaccurate data, regular expressions are used to detect them and be corrected manually and RE. For missing data, use the average value of the responses from the other students to fill in the missing value. Data after cleaning are standardized and uniform to allow intelligence and data analytics tools to easily access.

4.2.2 Database Structure

In order to realize quick access to school information, we use MYSQL to store school-related information. The following is the sample of school information database structure.



School_ID	School_Name	Location	Icon	Homepage	QSRank	TIMESRank	USNEWSRank
1	National University of Singapore	Singapore	http://nus.edu.sg/images/default-source/base/logo.png	nus.edu.sg	11	24	34
2	Kings College London	United King	https://upload.wikimedia.org/wikipedia/commons/thumb/	https://www.kcl.ac.Brita	31	35	37
3	The University of Warwick	United King	https://upload.wikimedia.org/wikipedia/commons/thumb/	https://warwick.ac.Brita	62	77	127
4	Nanyang Technological University	Singapore	https://upload.wikimedia.org/wikipedia/en/thumb/c/c6/Na	http://www.ntu.edu.sg/	13	47	43
5	University College London	United King	https://upload.wikimedia.org/wikipedia/en/thumb/d/d1/Ur	https://ucl.ac.Britain/	10	16	21
6	Technical University of Munich Asi	Singapore	https://upload.wikimedia.org/wikipedia/en/thumb/0/04/Te	https://tum-asia.edu.sg,	50	41	77
7	University of Edinburgh	United King	https://upload.wikimedia.org/wikipedia/en/thumb/3/3c/Ui	www.ed.ac.Britain	20	30	28
8	Eidgenossische Technische Hochs	Europe	https://upload.wikimedia.org/wikipedia/commons/thumb/	www.ethz.ch	6	14	25
9	Ecole Polytechnique Federale de L	Europe	https://upload.wikimedia.org/wikipedia/commons/thumb/	www.epfl.ch	14	43	45
10	The Chinese University of Hong Kc	HongKong	https://upload.wikimedia.org/wikipedia/en/thumb/8/87/Cl	www.cuHongKong.edu.h	43	56	113

Figure 4-4: Data Structure

5 Project Architecture & Implementation

5.1 Architecture Overview

The whole project is divided into 6 modules. Module 1 is the establishment stage of the data set. We use Python for website crawlers and set up our own database after some data preprocessing. The second module is the UI interface and the back end. After comparing the existing UI frameworks, we chose the VUE framework for writing the front end since the VUE framework implements the data binding of the response and the combined view components through the simplest possible API. As for the back end, we are using the currently most popular Django framework which is very easy to deploy. Module 3 is the Decision Table. We applied a rule-based method to conduct some specific processing of the input in this module. The reason for using rule-based technology was that these data were compared uniformly without many changes. Therefore, in order to obtain a faster processing speed, we finally chose the rule-based method. For the decision Tree module, the reason why we choose this model is that we hope that the model can be understood and explained can be visually analyzed, and can be easily extracted rules. The last module is the retrieval Model, we deploy Factorization machines in this module. After analyzing the data in our own dataset, we find that our data is very sparse since we have applied one-hot encoding for all the schools and majors existing in our database. Therefore, we need a model that can handle sparse data well. Therefore, we choose the FM model.

5.2 Process Flow

The illustration at the end of this section shows how the process moves from the UI interface to the back end, then to the decision tree service, then to the retrieval model, and finally back to the user.

1. Users fill in personal information on the UI platform. Any messages filled out will be forwarded to the back end as parameters
2. The back-end verifies the validity of the information according to the information filled in by the user. If the information is invalid, it returns an error to the UI interface and requires the user to fill in the information again; If valid, the message is unified and returned to the decision table.
3. According to different message types, the decision table searches for relevant rules and classifies them according to the rules. The decision table then presents the messages in the form of categories to the decision tree and retrieval model.
4. Based on the information given in the decision table, the decision tree obtains the final recommended school level, and finally returns the level to the retrieval model.
5. After obtaining the level of the recommended school, look up all schools and majors at that level in the database. For each major in each school, an input is formed along with the personal information returned from the decision table. With all inputs into the recall model, each major of
6. each school gets a score. Finally, the retrieval model returns a list of all schools, majors, and scores to the filter.
7. The filter filters the list returned by the recall model based on the preferred country of the user returned from the decision table, and removes schools from countries that the

user does not like. Finally, the user is given the top three schools and majors in order of their scores from highest to lowest.

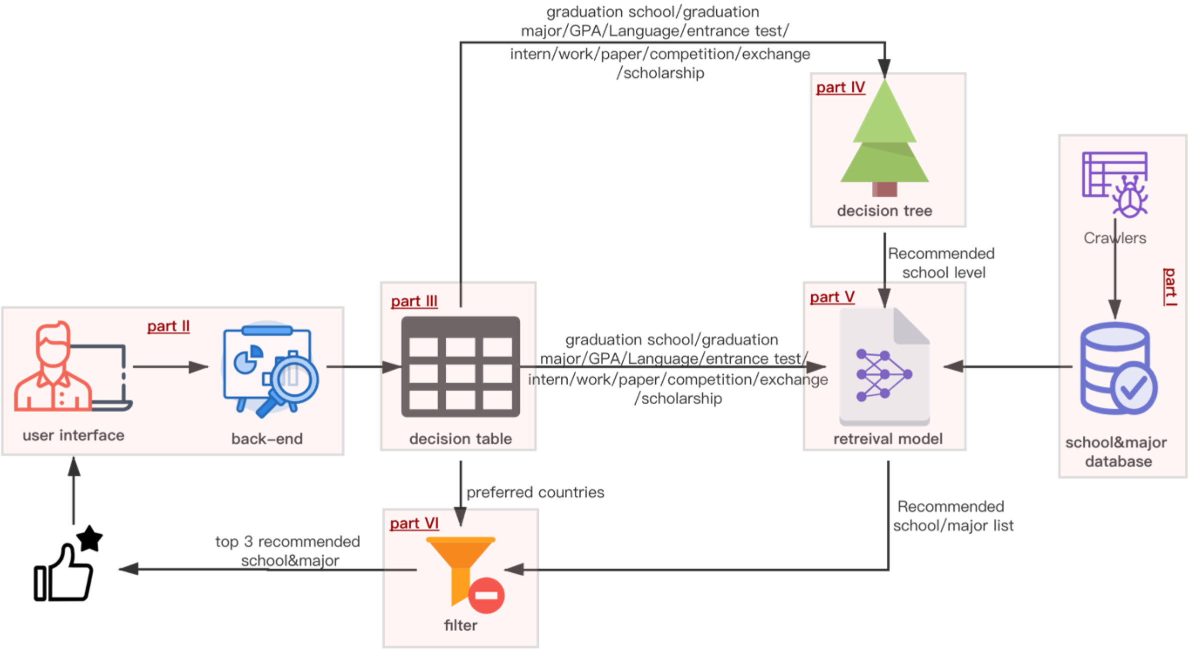

graduation school/graduation major/GPA/Language/entrance test/ intern/work/paper/competition/exchange /scholarship
part IV
decision tree
Crawlers
part I
Recommended school level
part II
part III
part V
user interface
back-end
decision table
graduation school/graduation major/GPA/Language/entrance test/ intern/work/paper/competition/exchange /scholarship
retreival model
school&major database
preferred countries
Recommended school/major list
part VI
top 3 recommended school&major
filter

Figure 5-1: Overall Architecture

5.3 System Modules

5.3.1 Decision Table

In this session, we will define a Decision Table, which has three main functions. First, it will encode the data from the Frontend, and map these attributes (such as GPA score, language score, entrance test score, etc.) into 5 or 2 categories (Table below). Second, it will pass these encode values into the Decision Tree, and received prediction results, i.e. target school grade from Decision Tree, then Decision Table will map the target school grade into S/A/B/C, and return it to Retrieval Model. Third, the Decision Table will calculate a LifeScore for each person based on the input attributes. This score will compare with the scores of the existing people in our database, and finally, get the percentage of people in our database the user has exceed. The calculation formula is:

$$\begin{aligned} LifeScore = {} & \text{school} * 10 + \text{gpa} * 6 + \text{language} * 3 + \text{intern} * 3 + \text{work} * 3 \\ & + \text{paper} * 4 + \text{competition} * 3 + \text{exchange} * 3 + \text{scholarship} * 2 \end{aligned}$$

Table 5-1: Corresponding table 1 of different attributes

Attribute	1	2	3	4	5
Language Rank(TOEFL)	>=100	[105, 110)	[95,105)	[85,95)	[0,85)
Language Rank(IELTS)	>=8.0	[7.5, 8.0)	[7, 7.5)	[6, 7)	[0, 6.0)
GPA score(100)	>=95	[90,95)	[85,90)	[80,85)	[0,80)
GPA score(4.0)	>=3.95	[3.81,3.95)	[3.58,3.81)	[3.25,3.85)	[0,3.25)
GPA score(5.0)	>=4.5	[4.0,4.5)	[3.5,4.0)	[3.0,3.5)	[0,3.0)
School	-	-	-	-	-
Entrance test(GMAT)	>=710	[680,710)	[640,680)	[580,640)	[0,580)
Entrance test(GRE)	>=330	[325,330)	[320,325)	[315,320)	[0,315)

Table 5-2: Corresponding table 2 of different attributes

Attribute	0	1
Intern	no	has
work	no	has
paper	no	has
exchange	no	has
scholarship	no	has
competition	no	has

5.3.2 Decision Tree

In Decision Tree, we use Python and Sklearn to do programming and apply machine learning techniques. After a few times adjustments to the Decision Tree, we finally obtain the best set of parameters for the model. The figure below shows our Final Decision Tree Model.

The feature that participates in the Decision Tree training phrase includes School, Language, GPA, intern, work, paper, competition, exchange, scholarship, and the labeled target school. Besides, we also use Grid Search and Cross-Validation Strategies to search for the best set of hyperparameters and validate models' performance, which boosts up robustness of the model.

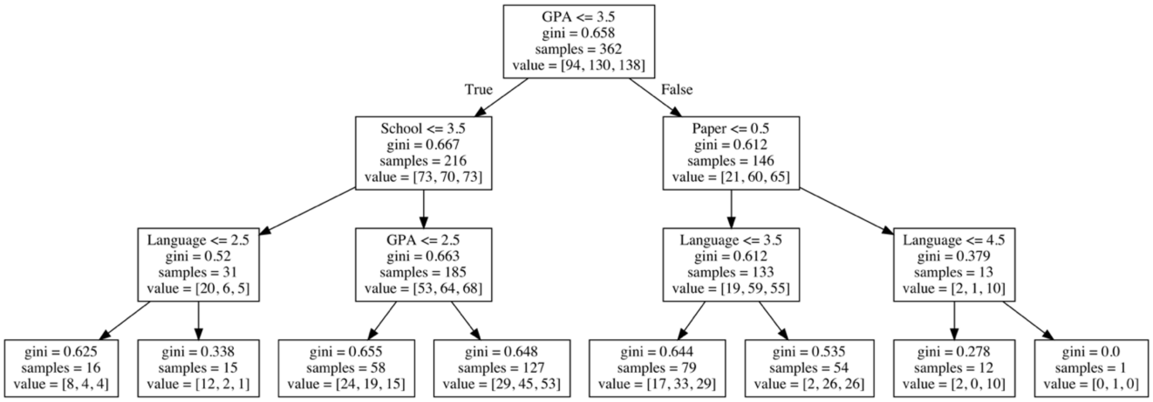


Figure 5-2: Decision Tree

5.3.3 Retrieval Model

5.3.3.1 Overview

The objective of this module is to use the retrieval model to recommend specific school and major recommendations for this user. The overall process is shown in the Figure 5-3: First, use the level of the recommended school obtained from the decision tree to obtain all the schools of that level and their corresponding majors in the database, and then use the personal information of the current users returned from decision table combined with each major of each school to form a piece of information, which serves as the input of the trained FM model. The final output of the FM model is the matching degree of the user with the school and major, and this information is returned to the next module.

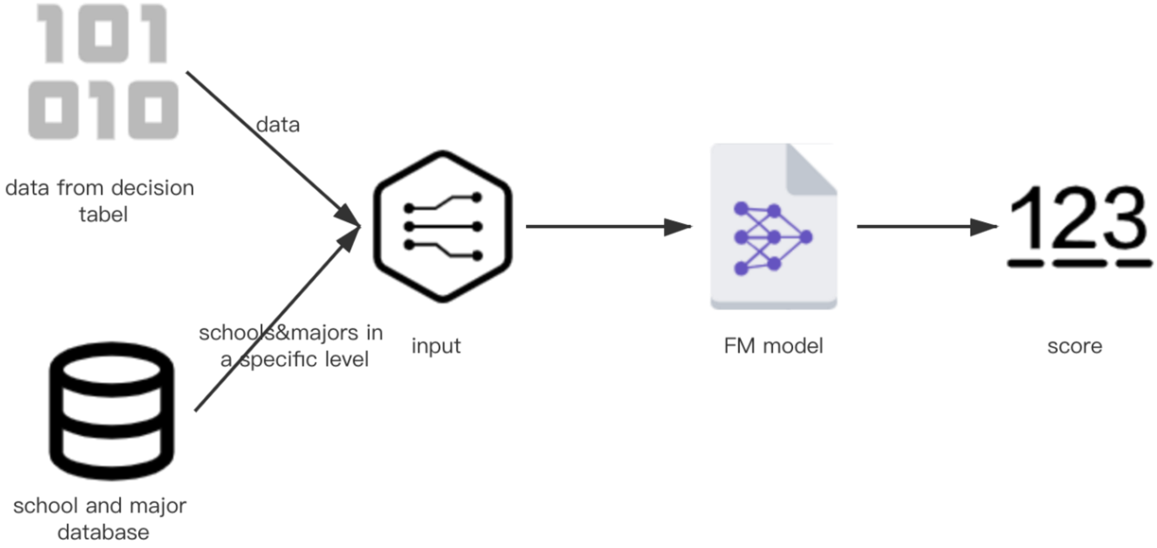


Figure 5-3: process flow of retrieval model

5.3.3.2 Model

In this module, we use Factorization machines (FM) as our retrieval model to get the specific recommendation considering that our data is very sparse.

Factorization machines (FM) [Rendle, 2010], proposed by Steffen Rendle in 2010, is a supervised algorithm that can be used for classification, regression, and ranking tasks. It quickly took notice and became a popular and impactful method for making predictions and recommendations. Particularly, it is a generalization of the linear regression model and the matrix factorization model. Moreover, it is reminiscent of support vector machines with a polynomial kernel.

Formally, let $x \in \mathbb{R}d$ denote the feature vectors of one sample, and y denote the corresponding label which can be real-valued label or class label such as binary class "click/non-click". The model for a factorization machine of degree two is defined as:

$$\hat{y}(X) = w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \quad < v_i, v_j > x_i x_j$$

where $w_0 \in \mathbb{R}$ is the global bias; $w \in R^n$ denotes the weights of the i-th variable; $v \in R^{n \times k}$ represents the feature embeddings; v_i represents the i-th row of $\mathbf{V}$; k is the dimensionality of latent factors; $\langle \cdot, \cdot \rangle$ is the dot product of two vectors. $< v_i, v_j >$ model the interaction between the i-th and j-th features. Some feature interactions can be easily understood so they can be designed by experts. However, most other feature interactions are hidden in data and difficult to identify. So modeling feature interactions automatically can greatly reduce the efforts in feature engineering. It is obvious that the first two terms correspond to the linear regression model and the last term is an extension of the matrix factorization model. If the feature i represents an item and the feature j represents a user, the third term is exactly the dot product between user and item embeddings.

5.3.3.3 Input and Output

The input and output of this module are shown in the Table 5-1. The input is the output from the decision table, including the school, major, GPA, Language, Intern, Work, Competition, exchange, scholarship. Meanwhile, specific schools and majors are also included (all schools and majors of that level are found in the database according to the grade of the recommended school obtained from the decision tree, and then one hot coding is performed for these schools and majors as the input). The output of this module is a score, namely the matching degree of a specific major of a specific school and the user.

Table 5-3: description of input and output of retrieval model

	VARIABLE NAME	DESCRIPTION
INPUT	School	Level of user's undergraduate school
	Major	Category of user’s undergraduate major
	GPA	Level of user’s GPA
	Language	Level of user’s language score
	Intern	Level of user’s intern experience
	Work	Level of user’s work experience
	Paper	Level of user’s paper
	Competition	Level of user’s competition experience
	Exchange	Level of user’s exchange experience
	Scholarship	Level of user’s scholarship
	School_name	One-hot encoding of recommend school name
	Special_name	One-hot encoding of recommend major name
OUTPUT	Result	A score indicating how matched this major of this school to the user

5.3.4 User interface

The Front-end Service is a Single-page Application (SPA) built with Vue framework, and the major components of UI are the checkbox, select, and input box. All use-cases of our application are built it. This figure below demonstrates the whole pipeline of one use case.

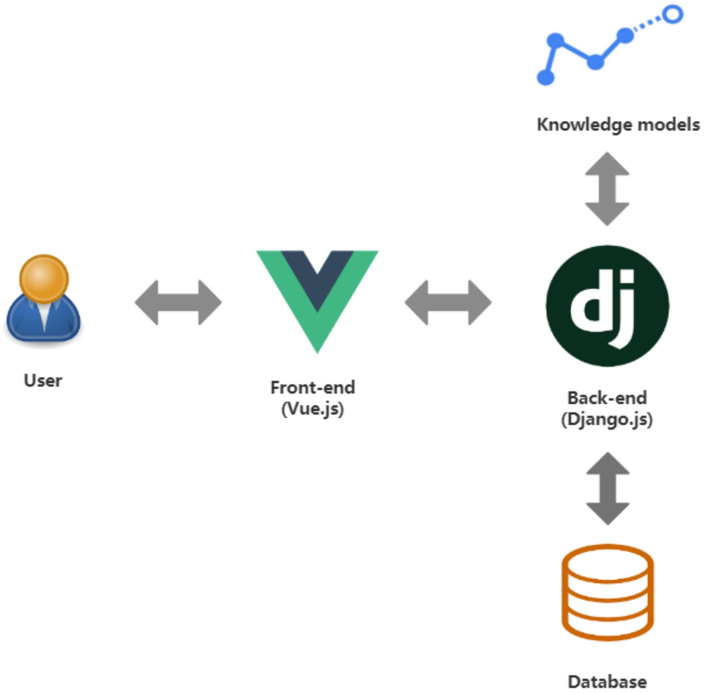


Figure 5-4: User Interface Pipeline

When a user completes the interested country page, the academic performance page, and the working experience page, then presses the submit button. This will trigger the Back-end Service Request API, which in-turn triggers various reasoning and functional requests to Rules Engine, Evaluation Service, and Recommendation Service. The table below shows all Request APIs.

Table 5-4: Request APIs

API URL	Description
/putUsersInfo	Pass all serialized data user fills in to back-end as the input in this system, including countries they are interested, academic performance, working experience.
/getEvaluateLevel	Back-end return the overall evaluate level according to the input information, then render it on our result page.
/getLevelDetail	Back-end return the each component of the evaluate level according to the input information, then render them on our result page.
/getRecommendSchool	Get Recommended school and major lists which are passed from back-end.
/getSchoolDetail	Pass the selected school name to back-end, then jump to the school detail page rendering this page with the detailed information from database.

Other miscellaneous functions in front-end UI include:

- Prompt the user which box is not filled in
- Check whether there is illegal input, such as GPA score can only fill in the number in the range
- Allow users to manually add and reduce table rows according to their needs
- Simulate transaction load in the process stream

For more details on user interaction with our application, please refer to the User Guide.

5.3.5 Backend

The Back-end Service is a single Python process built with the Django framework. Django itself is a high-level Python Web framework that bundles in a lot of clean, pragmatic designs influenced by experienced open-source communities, and therefore facilitates rapid development without the need to reinvent the wheel.

Our Back-end Service has the following architectural functions:

- Parse and pack data into each working component
- Integrate with Evaluation Service(with Decision Table and Decision Tree)
- Integrate with Recommendation Service(with Retrieval Model)
- Provide APIs for Front-end Service to consume (with Django REST)

6 Project Performance & Validation

In the process of creating a machine learning model, the goal is to achieve the best possible results. Our team's overseas study recommender is no exception because we hope that our retrieval model can recommend a suitable school based on the personal information of the end-user, and the more likely the school is to be admitted, the better. Therefore, we need to train the FM model with a set of training data. The data set of this model will be the input and output introduced in 5.3.3. The data set is divided into the training set and testing set. The training set is used to train the model and adjust the parameters. Next, the model is tested with a testing set to test the model's performance.

6.1 Dataset and Evaluation Matrix

6.1.1 Dataset

The data set we use to train the FM model in this section is our own crawled data. After feeding these data to the decision table, the same structure as above is obtained as the input of the entire model. As for the label, we defined the corresponding scores in the admission status column of the data set as shown in the Table 6-1: offer(offer with scholarship) is set to be 1.0, AD (that is, offers without scholarship) is set to be 0.9, and the status of waiting list is set to be 0.8. Reject is set to be 0.7 (this is because applicants only decide to apply for this school if they feel that they have a chance to be accepted, so we set it to a higher score instead of 0).

Table 6-1: Corresponding table of scores

Offer status	score	number
Offer	1.0	1089
Admission offer(AD)	0.9	518
Waiting List	0.8	51
Reject	0.7	485
Not apply	0.0	1399

To make the model more robust, we add some noise to the database. We think that the possibility of a particular person applying for a major that deviates greatly from his undergraduate major is very low. Imagine that a person studies liberal arts as an undergraduate, it is extremely unlikely that he will apply for a graduate degree in science. Even if he does, we think the probability of his being admitted is very small. Based on this fact, we add other categories of majors that are very different from his bachelor major of the school a person applies to into the data set as the user's noise, and the label is set to 0. In order to keep the data balanced, we randomly selected some people to add the noise. The distribution of the final dataset is shown in the third column of the Table 6-1.

6.1.2 Evaluation matrix

We use mean squared error(MSE) as our evaluation matrix to evaluate the performance of our FM model. In statistics, the MSE of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated. The formula is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widetilde{y_i})^2$$

where n is the number of samples, y_i is ground truth, and $\widetilde{y_i}$ is the output of the model.

6.2 Evaluation

In the experiment, in order to better test the performance of the model, we fixed K=15 (the dimension of auxiliary vector V of the FM model) and we have changed three parameters, which are learning rate, batch size and whether normalization is used, and the results are shown in the Table 6-2.

Table 6-2: performance matrix

Learning rate	Batch size	K	Normalization	MSE
0.001	16	15	Yes	**0.022**
0.001	16	15	No	0.024
0.001	32	15	Yes	0.037
0.001	64	15	Yes	0.036
0.0001	16	15	yes	0.042

6.2.1 Learning rate

Since the size of our dataset is not large and the problem that we want to solve is not very complex, the learning rate we chose is relatively small. We compared the performance of the model where the learning rate was 0.001 and 0.0001. The batch size was set to 4 and all the input data were normalized. The results are shown in the fourth and last columns of the Table 6-2. As we can see, for the same number of epochs, MSE is smaller when the learning rate is 0.001. We believe that the convergence speed is faster when the learning rate is 0.001. Therefore, in order to save training time, we use the setting of learning rate as 0.001 in our subsequent experiments.

6.2.2 Batch size

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. Since we did not use GPU for training and the size of our data set was not particularly small, we finally decided to use Mini-Batch for training the model. In order to evaluate the performance, we fixed the learning rate to 0.001, processed all the inputs in a normalized way, and then changed the batch size. We compared the batch size of 16,32 and 64 respectively, as shown in the second, fourth and fifth rows of the Table 6-2. As can be seen from the table, the best result was obtained when the batch size was equal to 16.

6.2.3 Normalization

Data normalization processing is a basic work of data mining, different evaluation index tend to have different dimensions and dimensional units, it will affect the result of data analysis. In order to eliminate the dimension influence between indicators, we need to normalize the data before feeding them into the model in order to solve the comparability among the data. After normalized processing of the original data, each index is in the same order of magnitude, which is suitable for comprehensive comparative evaluation. However, this is not always the case, and in some cases, the results are better without normalization. Therefore, we need to conduct trial and error to detect whether the normalization processing is needed. In this case, we fixed the learning rate to 0.001 and the batch size to 16, and compared the results of normalization (as shown in the first and second rows in the Table 6-2). The results show that there is no significant difference in the normalization of progress. Therefore, in order to save the time of data processing, we did not normalize the data in the end.

7 Challenge & Recommendation

Our team successfully deployed the system on the website, which can be accessed from the same local area network. Of course, this came with its own set of challenges and there are also many ways we can improve the implementation of the system with the luxury of more time and resources. The sections below describe the challenges we faced, how we tackled them and lastly further improvements for future releases.

7.1 Challenges

7.1.1 Data Acquisition via Web Crawling

As described in **4.2.1 Extraction Method**, information and data are extracted from China's largest study abroad sharing forum website through a Web crawling script. This website can better crawl and analyze scalar information, such as GPA scores and language scores because the requirements for filling in this information on this website are relatively clear and standardized. However, non-scalar information (or Additional Experience) is more complicated. All student awards, essays, and internships can only be entered in one text box, called other instructions. For the above reasons, it is more complicated to extract the features of this plain text narrative, and this may need some NLP knowledge which is kind challenging for team members.

In our data acquisition and cleaning, our team use RE to match the information separately, but the current data processing for Additional Experience only simply divides them into presence or absence. If using some NLP methods, it should be able to better analyze the gold content of the Additional Experience, so as to be divided into more categories. By doing so, data would be more diverse and this can improve our model's performance.

7.1.2 Reaching out to the Public

We believe that current study abroad information is open and transparent. If students only need school recommendation services, they will no longer need traditional study abroad agents as before. In addition, during the busy application period, overseas study agencies also need such assistants to help them better serve students. Therefore, this recommendation system service can satisfy the interests of students and institutions. However, for the system to achieve its goals, the first challenge is how to make it reach the public. This would mean for the system to be promoted and made known of its existence.

At this current stage in time, our team does not have the resources to market or promote the School Recommender in order for it to gain traction among a wider audience. However, we believe that word of mouth and the prevalence of social media would help us to boost the awareness of the recommender.

7.2 Future Improvements

7.2.1 Better Data Acquisition

Using Web Crawling to obtain data is vulnerable to website changing and data format inconsistency. Most of these data are raw, with inaccurate definitions and missing values. Also, the data we use has high redundancy and low coverage, thus, it is impossible to take users from all levels of schools into consideration. All these factors will directly affect the performance of the Decision Model. Therefore, how to obtain cleaner, more accurate, and wider coverage of public data is the next step we can do to improve the application.

7.2.2 Better and More Accurate Decision Models

As for the model, we tried Decision Tree and Random Forests, but the results are similar. Based on the consideration of data, we think that we can try other more specific machine learning models to fit these data. If we find the most suitable model, we will provide users with more accurate recommendation of schools to help boost up the success rate when applying schools.

APPENDIX OF REPORT A

Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal:

7 August 2020

Project Title:

ISS Project – School Recommender

Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)*

Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore
NATIONAL UNIVERSITY OF SINGAPORE (NUS)
Contact: Mr. GU ZHAN / Lecturer & Consultant
Telephone No.: 65-6516 8021
Email: zhan.gu@nus.edu.sg

Background/Aims/Objectives:

Main objective of this group project is to develop a school recommender system, which is capable of recommending appropriate school and major for students based on high-quality study aboard application cases data, providing professional assessment for the student's background and easy access to target school related information.

Requirements Overview:

- Research, data extraction & elicitation ability - from website, JSON
- Programming ability - Python
- Django framework understanding - apps, templates, statics, models
- System integration ability between Dialogflow and Django
- Clean data crawled from website
- Train and deploy decision tree model
- Train and deploy retrieval model

Resource Requirements (please list Hardware, Software and any other resources)

Hardware proposed for consideration:

- CPU

Software proposed for consideration:

- Django
- Vue
- Python
- Python libraries

- Django==3.1.1
- xlutils==2.0.0
- xlwt==1.3.0
- pandas==0.23.4
- xlrd==1.2.0
- openpyxl==2.4.11
- numpy==1.15.3
- ipdb==0.13.3
- demjson==2.2.4
- djangorestframework==3.12.1
- scikit_learn==0.23.2

Number of Learner Interns required: (Please specify their tasks if possible)

A team of 4 project members. Each bullet point represents one’s responsibility:

- Data Acquisition – crawl aboard application data, Data Processing, User Interface development, Django & Database development
- Product prototype design – product solution, business flow design, UI design, System architecture design – system/data flow design, system modules design, database structure design, Data Processing, User Interface development, Django & Database development
- System architecture design, Data Acquisition, Data Processing , Retrieval model development – model training/hyperparameter tuning/model evaluation, User Interface Development
- Data Processing, Decision Table – expert rule survey and design, module development, Decision Tree development

Methods and Standards:

Procedures	**Objective**	**Key Activities**
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of the project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	To develop the source code in accordance to the design. To perform sub-model testing to ensure the quality before sub-models are integrated as a whole project	1. Setup required development environment 2. Understand the System Context, design and system dataflow 3. Split requirements into sub-models. 4. Perform coding, identify and introduce tools /libraries to meet system’s requirement
Integration with different platforms and services	To ensure interface compatibility and confirm that the integrated system software meets requirements.	1. Prepare system test specifications. 2. Conduct system integration testing 3. Evaluate testing
Acceptance Testing		1. Plan for Acceptance Testing

	To obtain ISS user acceptance that the system meets the requirements.	2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guideline must be provided in ISS production (ISS standalone server) format 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: Group 4
Project Title (repeated): ISS Project –School Recommender
System Name (if decided):
Team Member 1 Name: Lin Xi
Team Member 1 Matriculation Number: A0215403W
Team Member 1 Contact (Mobile/Email): +65-88520759 / linxi@u.nus.edu
Team Member 2 Name: Liu Chenxi
Team Member 2 Matriculation Number: A0215461M
Team Member 2 Contact (Mobile/Email): +65-82438399 / E0535551@u.nus.edu
Team Member 3 Name: Cao Wen
Team Member 3 Matriculation Number: A0215516L
Team Member 3 Contact (Mobile/Email): +65-88896411 / caowen@u.nus.edu
Team Member 4 Name: Zheng Kai

Team Member 4 Matriculation Number: A0215414R
Team Member 4 Contact (Mobile/Email): +86-15968822493 / kzheng@u.nus.edu

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		

Advisor Assigned:

Contact: Mr. GU ZHAN / Lecturer & Consultant
Telephone No.: 65-6516 8021
Email: zhan.gu@nus.edu.sg

APPENDIX OF REPORT B

Mapped System Functionalities against knowledge, techniques and skills of modular courses

Modular Courses	System Functionalities / Techniqe Applied
Machine Reasoning (MR)	• **Knowledge Elicitation and extraction**: Web crawling from websites & repositories, manual extraction from websites/internet • **Knowledge Representation**: Decision table for students' applicant records such as target school level, GPA level and language level, data representation for django database (school's information as example explained in Section 4.2), process flowchart for architecture system, process flowchart for retrieval model. • **Rule Based System**: Decision tree to derive rules (expressed in tree structure) for recommending the most suitable school level
Reasoning System (RS)	• **Search & Optimization**: Genetic Algorithm search technique applied on decision tree to optimize the results.
Cognitive System (CGS)	• **Cognitive System**: Retrieval Model to recommend specific school and major recommendations for this user

APPENDIX OF REPORT C

Installation and User Guide
(Refer to separate document for Application & Deployment User Manual)

APPENDIX OF REPORT E

Individual Reports

Individual Report: Cao Wen (A0215516L)

Personal Contribution

Throughout the project, I worked as backend developer to build the bridge between users' feature input and predict outcomes. Aside from that, I was also in charge of user information pages coding in frontend. After collecting users academic performance and additional experiences, I had drafted out the rule-based decision table for the most suitable school level and implemented it into the backend code using Python in Django framework.

I also participated in using feature engineering technique to extract and engineered useful information from the student's application records detail as predictors, then using scikit learn decision tree library to build prediction model and few rounds of iteration to make sure the model can obtain high accuracy performance before output the prediction into decision tree rules.

Besides, I had created web scrapping scripts to extract important data from one of the largest study abroad forum website in China. Furthermore, I have been collaborating closely with my teammates during the implementation and debugging process of the project. As for the report, I was involved in Section 4,5,7 in data acquisition, user interface, backend part.

Learning outcome

Throughout the module and the group project, I have learnt so many new things. First is definitely how to integrate a intelligent recommend system using Vue framework, Django framework, Python modules. In my past learning journey, I did not have the experience in developing a complete intelligent product. To be specific, I trained and tested a predict model on the dataset, but did not have an accessible way to present this model to everyone, especially for those who do not know programming. After this project, I can actually build an intelligent product since I am familiar with the pipeline.

Other than that, this project has taught me the challenging part of real-life data, where developing data cleaning algorithm can consume half of my time in this project. The noise present in real life data is unpredictable and can be in any forms, example typo error, wrong information given inside the application detail or missing information. This has motivated me to design a scalable or generalize algorithm which can be re-used for different condition.

Last but not least, I have found Github is an efficient way to collaborating with teammates in term of software development. By creating branch in Github for testing new functions of our system, the project workflow will be more systematic and efficient, without blocking the workflow of other teammates.

Knowledge and Skill Application

Recommender systems are among the most popular applications of data science today. Amazon uses it to suggest products to customers, YouTube uses it to decide which video to play next on autoplay, and Facebook uses it to recommend pages to like and people to follow. They are among the most powerful machine learning systems that online retailers implement in order to drive sales.

With the recommender development knowledge that I have gained, I would like to apply the knowledge in helping myself, family members or even friends, if anyone need some advice to choosing the most suitable one in any field, and they do not want to spend time in searching every webpages to find the answer, plus experts and agencies are expensive. Recommender system can help with it. First is the data collection phase where data can be classified either as explicit or implicit, then the training phase applies mathematical algorithms and statistical analysis on the collected data to “learn” the patterns that are present.

Further to this, Django is also one of the important lessons I learned, it can be widely used in many web applications. One of Django's main goals is to simplify work for developers, it uses the principles of rapid development, which means developers can do more than one iteration at a time without starting the whole schedule from scratch. Being able to deploy web applications is very useful for many real life, using Django allow a developer to turn his idea into a complete product much faster. With this experience, I can have greater confidence in further deploying bigger web applications in the future.

Individual Report: Lin Xi (A0215403W)

Personal Contribution

In this project, my main contribution has six parts. The first one is the project management, which includes project planning, deciding scope of our application, scheduling of tasks and events for each team member. The Second one is the product prototype design, mainly including product solution, business flow and UI design. The third one is contributing to the system architecture design, not only I have participated in defining the system modules and data flow but also I have design the database structure. The Fourth part is the front-end user interface develop, I have responsible for the basic configuration of the front-end VUE framework, and the development of index, recommend report, school detail pages development and their communication with backend service. My fifth contribution is to design the structure of our knowledge base, and the raw data preprocessing strategy. Last but not least, I completed the basic configuration of Django backend framework including the model configuration and the communication with frontend, and get school detail information API.

At the very beginning of this project, I contributed to the project idea generation by leading my team members to provide project proposals. The project proposal includes three questions, the first one is to describe a existing problem in our daily life or business world, the second one is to briefly provide your solution to solve this problem, last one is the technology and data you need to implement this solution. Thus after we compare our proposals from business value, project feasibility etc., we decide the project idea. We decide to develop web application for this project. Based on the decision , I designed the product modules and its business flow, decided the scope of our application which is important for our system architecture design and project management. Based on our project objective to find out the suitable target school depending on user's undergraduate background, I have designed our raw data preprocessing strategy. Expert rule was introduced to make continuous data categorical, and I used python regular expression to extract key information and format data from description which is in natural language. In frontend development, I built up the basic VUE framework for our group, and developed parts of the pages. In addition, I have helped my teammates adjusting layout for their pages. As the main backend contributor, I built up the basic Django framework and database, coding functions to link the user input data with algorithm models, and retrieve data from database return to user frontend.

Aside all the main responsibility in the project, I have also helped my teammates in optimization of the decision tree and FM retrieval model performance by providing data sampling solution, data pre-processing, post-processing solution and hyperparameter tuning solutions.

Learning outcome

Through the group project I have learned a lot in project management, web data scraping, decision tree, recommendation system and VUE, Django Framework.

First of all, in this project, I have learned setting clear project goal in each project phase is very important in pushing forward the project. For example, in the business survey and project idea generation phase, our goal is to generate an idea has business value and can be accomplished with the limited time and resource, in the development phase, our goal is to finish each module depends on the product prototype and system architecture design. A clear project goal ensure our project resources are devoted to the most important things, each team member is clear their own task boundary and the resource they will need.

Second, I have learned using python package to do web data scraping. I have become more familiar with the webpage html structure when try to scrape the data we need.

Third, decision tree. In the system model design phase, I have learn more about decision tree's traits, like easy to understand and interpret, perfect for visual representation. The majority of our feature is numerical or categorical, thus I decide use decision tree as the tool to evaluate the overall level of a user based on his/her background information. With the help of decision tree, we can easily discover which feature of the student is the most important in evaluate his/her background. In model training phase, the decision is easy to overfitting, I have learned limiting its depth to avoid overfitting.

Fourth, VUE and frontend development. In the project, I have learned the use of VUE framework, familiar with the VUE lifecycle, the relationship between different VUE components, router and call API with the help of axios.

Last but not least, the design of recommendation MVP. I have learned the requirements of the data, algorithm, output, system modules for a minimum recommendation system. Understanding the idea of recommendation system to make use of history user data to do recommend for future users.

Knowledge and Skill Application

As mentioned above, In the project, I have experienced the whole phase of from business survey, designing the recommendation product solution and prototype, the system modules, data flow, and recommend result visualization. These knowledge and experience I have learned will definitely help me in future when implementing recommendation system for other business scenarios.

First, I can apply this experience into formatting the business data into knowledge base according to different business needs. And make use of these knowledge base to assist people in business operation.

Second, with the help of recommendation system, the user data can be used to recommend products or service for similar user to promote the sales.

Last but not least, the VUE and Django framework I learned enable me to do fast and easily development and deploy of front-back separation web application which is runnable on varieties of devices with different OS.

Individual Report: Liu Chenxi (A0215461M)

Personal Contribution

In the whole team project, I mainly trained as an algorithm developer and tested the retrieval model to recommend specific schools and majors based on users' personal information. At the very beginning, I participated in the design of the entire system architecture, including data flow design and system module design. During the data crawling phase, I crawled the data from website to further establish our own data set. As for the data pre-processing stage, I was responsible for the processing of undergraduate schools and undergraduate majors in the original crawled data. In this process, I de-duplicated the data, unified the format, checked whether the data was valid and classified them. In the retrieval model training stage, I conducted model selection, hyperparameter selection and model testing. To increase the robustness of the model, I also did data augmentation (adding noise). When writing the UI interface, I helped write the page for country selection. In addition, I have been working closely with my teammates in the implementation and debugging of the project intention. As for the report, I have completed part 5 of the overall project architecture, the retrieval model, and Part 6. Last but not least, I played a part in the video presentation of the group project.

Learning outcome

The most useful lesson I learned through this project was how to use the retrieval model for different recommendations according to different person. I've learned that we can think of this problem as a regression problem, taking personal information and specific recommendations together as input, and then use model to calculating how well they match. The higher the final score, the higher the match. I also learned about the specific retrieval model, the FM model, which not only learned the relationship between input and output, but also learned the relationship between the different features of input data. At the same time, I learned that the FM model had a good performance for dealing with sparse data sets because it introduced an auxiliary vector V. In addition, I learned how to augment the data to increase the robustness of the model. I've learned that sometimes we need to artificially add some noise to make the model more expressive.

During this project, I also learned how to design the web by writing the front end using the VUE framework. I learned the relationship between HTML, CSS, JS and their applications. Besides, through this project, I mastered how to collect user input from the front end, and then pass that input into the back end (which handles requests and manages user databases) in the JSON-formatted as a request.

In addition, I learned how to crawl website data using Python. Since what we crawl is the data of forum, the data format is not unified. Therefore, I also learned how to use regular expressions to do some filtering on the data through data pre-processing. Meanwhile, I learned and became familiar with the use of the Pandas library and had a certain understanding of Python data processing.

During the final joint debugging process, I learned how to collaborate effectively with my teammates when developing software on Github. By creating branches on Github to manage different modules, the project workflow will be more systematic and efficient, without blocking the workflow of other team members.

Knowledge and Skill Application

The knowledge and skills gained through this project will definitely be very useful in future jobs. As mentioned above, how to make personalized recommendations is an important experience I have gained. Recommendation systems can be widely used in many industrial fields. Being able to make different recommendations for different users is useful for many products. For example, music software can recommend suitable songs to users according to their listening track. Other examples include online shopping sites such as Taobao and Shopee, which make recommendations based on users' personal information. Therefore, being able to understand how to write a recommendation system will open many doors and jobs and is an important skill in IT.

In addition, writing web pages using the VUE framework and Django was another important lessons I learned. Web applications are needed in almost every company, and Django can be widely used in many Web applications. Being able to deploy Web applications is useful for many real-world purposes.

Finally, learning how to use pandas library to process data in this project will also be of great help to me. In today's society, we have a large amount of data, how to do some processing of these data is an important issue in the IT field. Therefore, learning how to batch process data is very helpful for future work.

Individual Report: ZHENG Kai (A0215414R)

Personal Contribution

In the entire team project, I proposed the idea of the development of recommended projects for studying abroad, and worked out with the team how to apply the knowledge learned in the Intelligent Reasoning System to the project. I participated in the process of data cleaning and pre-processing, including the grading and classification of target colleges and universities, and the grading of examination languages. In the development of the decision-making module, I tried the interpretable decision tree model which is commonly used in machine learning to process the characteristics of the user's overseas study data, and finally let the model predict the user's target college level. At the same time, in training the model of the decision tree, I carried out a series of feature engineering, which is to obtain the most beneficial features for the model from a large number of features to train the model. In the end, the model I trained can achieve satisfactory accuracy on our artificially divided validation set. At the same time, I also participated in organizing the prediction of the model into an interface as a sub-module and merge the code into the main program. Therefore, I am one of the core algorithm development members of the entire project. In addition, I also completed the writing of the algorithm part of the report.

Learning outcome

In the project, I learned how to do data collection, data cleaning, and encoding the data into that is accessible for the model. Through this process, I understood the difficulty and complexity of data collection. In order to achieve the purpose of classifying overseas study data, I tried a variety of feature combinations, and tried a variety of machine learning algorithms, and found the gap between theory and facts. In real or noisy data, many models do not perform as well as the paper said. So how to deal with these data will be the first consideration for algorithm engineers. At the same time, in order to conduct scientific experiments, we need to formulate rigorous training strategies and standardized data division strategies.

In teamwork, the readability of the code is one thing that every member needs to consider, so how to write the code in a clean and tidy way is another benefit that this project has taught me.

Knowledge and Skill Application

The knowledge and skills gained through this project will greatly inspire my future work. How to apply machine learning to real or noisy data is the most crucial thing that an algorithm engineer must consider, and it is also a necessary requirement to test the qualification of an algorithm engineer. In addition, how to use Git to achieve efficient teamwork and use python to write clean code is another thing I learned from this project.