COMP4801 Final Year Project

Mining HKUCS Graduate Student Data: Extraction, Analysis, and Prediction

Final Report

Supervisor: Dr. Reynold Cheng

Group Member:
  Wu You (3035086120)
  Xu Fangyuan (3035085530)

Author: Wu You
Submission Date: 16th Apr 2017
DISCLAIMER

Information published on the web application of this project is provided for educational purposes only. The Computer Science Department of the University of Hong Kong and the project team is committed to ensuring that the application data of our applicants for graduate studies are treated with respect and safeguarded to ensure privacy.

ABSTRACT

Educational Data Mining (EDM) is an emerging field that uses statistical and machine learning algorithms to find patterns in massive educational data sets. It is commonly used to solve problems such as predicting admission outcome and student performance. This project applies educational data mining techniques to the admission data of the graduate programs in the Computer Science Department of the University of Hong Kong (HKUCS). The objective of the project is to build a web application with a user-friendly interface that visualizes and analyses admission data using the following algorithms: Decision Tree algorithm, Association Rule algorithm and Logistic Regression algorithm. Based on the admission patterns that are found, the web application suggests a ranked list of candidates for admission decisions.

ACKNOWLEDGEMENT

The project would not have been completed without the kind support and help of different individuals and organizations. We would like to express our gratitude to all of them.
We would like to express our thanks to Dr. Reynold Cheng, our supervisor, for providing useful guidance and feedbacks for the design and methodology of our project. His continuous support throughout the project is extremely valuable to this project.

We would also like to express our gratitude to Mr. Jiafeng Hu, who has extensive experience in the admission process of the graduate programs in HKUCS, for his time and efforts to help us collect admission data and providing us with his useful suggestions on this project.
Table of Contents

DISCLAIMER ........................................................................................................................................... 1
ABSTRACT ................................................................................................................................................ 1
ACKNOWLEDGEMENT ........................................................................................................................... 1
INTRODUCTION ........................................................................................................................................ 4
PROJECT BACKGROUND ...................................................................................................................... 6
RELATED STUDIES ................................................................................................................................ 7
PROJECT SCOPE ...................................................................................................................................... 8
   Extraction Layer .................................................................................................................................. 8
   Analysis Layer .................................................................................................................................... 8
   Prediction Layer .................................................................................................................................. 9
DELIVERABLE ........................................................................................................................................... 10
METHODOLOGY ...................................................................................................................................... 12
   Technology Used ................................................................................................................................ 12
      Front-end Development ...................................................................................................................... 12
      Back-end Development ...................................................................................................................... 13
   System Structure ................................................................................................................................. 14
      Functional Framework ....................................................................................................................... 14
      Database Design ............................................................................................................................... 16
FUNCTIONALITIES AND IMPLEMENTATION ...................................................................................... 17
   Extraction ........................................................................................................................................... 17
      Data Collection ................................................................................................................................. 17
      Data Pre-processing .......................................................................................................................... 17
   Visualization ....................................................................................................................................... 19
      Dashboard Visualization .................................................................................................................. 20
      Applicant Detail Visualization ......................................................................................................... 22
      Customized Visualization ................................................................................................................ 23
   Analysis ............................................................................................................................................... 24
      Decision Tree ................................................................................................................................. 24
      Association Rule .............................................................................................................................. 26
INTRODUCTION

Educational data mining (EDM) is a growing research area that aims at studying educational problems such as student admission and curriculum evaluation, by exploring data originating from educational context. [1] It is generally acknowledged that substantial valuable information, which could be vital to many educational decisions, is embedded in the educational data. Unfortunately, EDM is still not playing a part in the admission decision-making process in the Computer Science department of the University of Hong Kong (HKUCS). Each year, HKUCS attracts hundreds of applications from students all over the world for doing a graduate study. From these applications, numerous data have been
collected, including student profiles, academic performance, research interests, scholarships, teachers’ comments and admission decisions. However, these data have not been well utilized due to the complexity of the large range of data to be analyzed. Firstly, parameters of different types, such as applicant’s GPA, applicant’s English skills and applicant’s research interests, correlate with each other and they all contribute to the admission decision-making. The cross correlation among the data leads to increasing difficulties in analyzing the importance of each parameter and unraveling their relationships. Moreover, for the same parameter of two samples, there might be different interpretations of the data with the same value. For example, two applicants with GPA equal to 3.6 might have different academic performance due to the difference in grading policies. Therefore, to address the above problems, an EDM tool is needed so that the administrators and teachers of HKUCS can mine the candidates’ data and probe for significant patterns or trends that are crucial to graduate student admission.

The goal of this project is to develop an educational data mining web application, named HKUEDM, which can make use of the graduate students’ data to conduct data extraction, visualization and analysis. Moreover, based on the analysis, HKUEDM can build classifiers to classifier applicants into different categories and predict their admission results. With these functionalities, HKUEDM can help users discover important patterns or trends that provide insights into the admission strategies as well as facilitate the admission process of HKUCS.

The final report proceeds as follows. It first introduces the project background and the shortcomings in the existing online admission system. Then it examines the previous studies conducted in the field of Educational Data Mining and what insights their findings have provided to this project. It also outlines the scope and deliverable of this project. Following that it discusses the methodology and technology used to develop HKUEDM. It then
introduces the overall system structure and the detailed implementation of each function. Furthermore, it presents the results of the data mining analysis on the admission data and their implications. In the end, it discusses the possible directions for future work.

PROJECT BACKGROUND

Before the project, there exists an online admission system. After investigation into this admission system, we found the following problems. Firstly, it is rather difficult to navigate through a long list of applicants, which makes it inconvenient to identify and locate a particular applicant. Secondly, the user interface is not user friendly. For instance, it is difficult to evaluate an applicant given all his or her information presented in a paper-like format. Furthermore, the existing visualization of data in the system is very limited and only visualizes the comparisons based on a single attribute. Last but not least, it only provides a graph of a decision tree without any interpretation and users cannot interact with the system to generate customized mining results.

After identifying the above problems, we conducted interviews with Dr. Reynold Cheng, the head of graduate admission in HKUCS, and Mr. Jiafeng Hu, who is also highly involved in the admission process to better understand the requirements of HKUEDM and potential user case. After deriving better understanding of the user requirement, we found that the evaluation of applications in the initial stage of the admission process is very time-consuming for professors who are already busy with their own schedules. Thus, it would be very helpful if HKUEDM can present various forms of visualization, provide different mining tool for user to conduct mining analysis on their interested data, and make sensible prediction and recommendation based on the mining analysis results. Therefore, we decided to divide the project into three layers, extraction (visualization), analysis and prediction.
RELATED STUDIES

In general, the most commonly used techniques in Educational Data Mining include prediction, clustering and relationship mining. [1] We reviewed the following existing projects that used these techniques to build data mining tools and produced significant results.

Mashat et al. demonstrated how association rule can play a role in EDM. They chose to use the Apriori algorithm, one of the association rule mining algorithms, on the university admission to extract relations between various attributes like students’ exam results and courses taken by the students. [2] These relations can be used to compare the importance of different assessment criteria during the admission process. In this project, we also include the Apriori algorithm in association rule mining.

Feng et al. utilized the Self Organizing Map (SOM) neural network, cluster analysis and Fayyad data mining model to establish a university admissions decision-making model. [3] The model includes the geographical data as an important attribute of applicant and develops a new reference for admission scheme and propaganda. [3] However, since the data size of this project is too small to be applied on the SOM neural network, their model is not adaptable in this project.

Fong et al. designed a recommender system by use of Decision Tree classifier for university admission in Macau. They implemented a back propagation algorithm to train a learning model that computes a weighted importance of different attributes. Then they applied the Decision Tree algorithm to build the classifier of the recommender system by adding the most important attribute every time till the error rate of the decision tree stops to decrease. [4] Their model provides insight of the use of Decision Tree classifier in EDM and we also chose to implement a Decision Tree classifier.
PROJECT SCOPE

Given the complexity of the problem, the scope of the project is divided into three layers.

Extraction Layer

The first layer is to identify the useful data based on users’ requirements and present them in an understandable way through different forms visualization. The major two forms of visualization are introduced as follows.

1. Visualization on a set of applicants includes various types of charts, such as bubble chart, pie chart and line chart, so as to provides user with an overview of the trend and pattern underlying the admission data. HKUEDM generates charts that concern multiple attributes so that it is easier for users to visually compare and evaluate the applicants. Moreover, HKUEDM also provides customized visualizations, through which the set of applicants, the set of attributes and the type of customization can be customized. In addition, the visualization also has features like using mouse-over to display further information so as to provide more kinds of interactive functions and help users visualize the data in a more convenient way.

2. Visualization on a specific applicant presents his or her application data in a user-friendly way by highlighting the important information and adding comparison data to help users learn about the applicant’s performance in each attribute compared with other applicants.

Analysis Layer

In this layer, HKUEDM provides users with a set of data mining methods to do analysis in the application data. It uses Decision Tree algorithm and Association Rule algorithm to
support the functionalities provided in this layer. On top of that, it creates a user friendly interface for the users to easily apply the algorithm to selected set of data to conduct customized analysis. By default, HKUEDM runs the data mining algorithms with a predefined set of input and generates a model with the most satisfying result. The result is visualized in a simple and direct way so that users do not need prior knowledge to learn some insights from the data with such a one-click approach. Other than this simple one-click approach, HKUEDM also allows users to conduct customized analysis by selecting the range of data to be used, the set of attributes to be included and the values of each parameters. The detailed functions provided by each algorithm in this layer are listed as follows:

- **Decision Tree.** This function allows the user to choose a set of attribute to be classified. Users can also specify the set of data to be analyzed by selecting the year of applicants to be included. It presents the decision tree in a tree graph, which forms the decision rules for classifying applicants. It also displays a ranked list of attributes based on their importance.

- **Association Rule.** This function first allows users to select a set of attributes and a threshold value for each attribute so as to transform continuous variables to binary variables. Then it considers attributes as items and process each applicant’s data to generate a transaction. In the end, it applies the Apriori Algorithm, which uses a breadth-first search strategy to generate insightful rules and relations underlying the data with statistical measures such as **Support** and **Confidence** for user’s reference.

**Prediction Layer**

The web tool implements Logistic Regression algorithm so as to predict applicants’ admission results. Combined with the patterns found in Analysis Layer, the web tool
processes the incoming application data and classifies the applicants into different categories to facilitate the admission decisions. The functions provided are listed as follows:

- **Logistic Regression.** This function first allows users to select the year of applicants to serve as testing data and the target categories to serve as dependent variable. Then it uses all the other years’ applicant data to serve as training data and build the Logistic Regression classifier. Then it uses this classifier to predict the probability for each category for each applicant in the selected year. In the end, it ranks all the applicants according to the probability for each category and presents the results to the users.

- **Outcome Prediction.** This function makes use of the results of *Logistic Regression* and *Decision Tree* and suggests a ranked list of applicants to users as potential candidates for short-listed interview, potential candidates for a program and potential candidates who will receive scholarships like Hong Kong PhD Fellowship (HKPF).

**DELIVERABLE**

The deliverable of the project is a web application (HKUEDM). HKUEDM provides a user-friendly interface (See Figure 1) with a set of functionalities specified in the *Project Scope* session.
Figure 1: User interface of HKUEDM
METHODOLOGY

Technology Used

Django is chosen as the web framework for developing the HKUEDM, which follows the model-view-template (MVT) architectural pattern. The primary reason for choosing Django is that Django’s emphasis on reusability and scalability of components can greatly ease the work of building this complex, database-driven web application and provide better security [5]. Moreover, as a time-tested and crowd-tested framework with detailed documentation, Django has high quality and we can easily find solutions when encounter difficulties during development through the large and supportive Django community. The implementation is then divided into front-end and back-end development.

Front-end Development

The main responsibility of front-end development is to implement the data visualization and interactive functions. To help better realize these functionalities, one of the existing libraries we used is d3.js, a JavaScript library that combines powerful visualization components and a data-driven approach to Document Object Model. As one of the most widely used JavaScript libraries for producing dynamic, interactive data visualization, d3.js provides numerous types of visualizations and interactive functions. d3.js not only provides a set of easy-to-use APIs, but also allows substantial flexibilities for users to customize the visualization with their own designs. Despite of the steep learning curve, there are extensive examples and resources available online for us to study and adapt.

In addition, other visualization libraries such as morris.js and chart.js are also used in this project.
Back-end Development

- Server: Currently, HKUEDM is running on a local server. We will try to host the web application on a hosting server in the next stage.

- Database: MySQL is chosen since it is free and open source. Although it is lightweight, it provides all the features that are needed in this project. (https://www.mysql.com/why-mysql/) Moreover, with a wide user base, it is easier to find support and solutions when there occurs unexpected problem.

- Application:
  - Programming language: Python is chosen as the programming language, which handles interaction with the database and implementation of the data mining tools and the classifier. We have chosen Python over R, which is a language focusing on data and statistical analysis for the reason that Python is more readable and simpler and thus has a relatively low learning curve, which is more suitable given the limited time frame of this project. Apart from that, Python is also widely used for data mining and has a large community. Thus, there are a lot of data mining libraries that implement a large range of data mining algorithms that suit our needs.
  - Data mining methods: Among the existing data mining libraries for Python, we chose Scikit-learn for the following two reasons. Firstly, it consists of series of data mining modules that cover the full scale of common data mining process, including data pre-processing, classification and prediction. Moreover, it can be used to interoperate with other libraries like visualization libraries and scientific computing libraries. The libraries used along with scikit-learn in the project are listed as follows:
- Numpy is an extension to the Python programming language, adding support for large, multidimensional arrays and matrices as well as high-level mathematical functions. Since the admission data contains large range of attributes, Numpy is very useful in processing these data.

- SciPy is an open source Python library used for scientific computing. Its modules for clustering, spatial data manipulation and optimization and functions like splitting data-sets is very useful for data analysis.

- GraphViz is a package for drawing graphs. It is used to visualize the results generated in the data analysis.

**System Structure**

**Functional Framework**

Figure 2 illustrates the framework of the functionalities of HKUEDM. The framework is mainly formed by the three layers: Extraction Layer, Analysis Layer and Prediction Layer. The figure also introduces the functions implemented as well as their major sub-functions for each layer. The communication between layers and the database are also included.
Figure 2: The functional framework of the system
Database Design

Figure 3: Class Diagram for database

The figure 3 depicts the database design, which comprises the major table, applicants, and other Django tables that handle authentication and administrative work. The applicants table consists of 45 columns, which refers to 45 attributes of an applicant data.
FUNCTIONALITIES AND IMPLEMENTATION

Extraction

Data Collection

In September, 2016, we collected the admission data of 2016 applicants. In January, 2016, we collected the admission data of 2014 and 2015 applicants. In February, 2017, we collected the most updated ranking data for applicant’s previous educational institutions by using the QS World University Rankings by Subject 2016 - Computer Science & Information Systems [6]. In March, 2017, we collected the admission data of 2017 applicants.

Data Pre-processing

After we gathered the admission data of the past four years, we found that there exist some problems in the data, including out-of-range values, missing values, inconsistent data scale, incorrect spelling, inconsistent data format. Since analyzing data that has not been carefully screened for such problems can produce misleading results, data pre-processing is carried out for all the data collected.

We first pre-processed the admission data that contains attributes of different forms or scales and converted them into a uniform format or scale. For numerical data of different scales, such as GPA and QS Ranking for universities, we used standardization methods to convert them into values that are comparable. For categorical attributes such as research interests and awards, we encoded them into numerical values. For example, we added a shortlisted binary attribute to indicate whether a student has been shortlisted for an on-site interview.

Then we pre-processed the admission data that have messy formats to extract the useful information. For example, the admission data combine the applicant’s different English tests
results to form a single attribute named *english_tests*. We parse the data so as to extract applicant’s test result for TOEFL, GRE, IELTS and CET6 separately.

Moreover, we corrected some of admission data that have incorrect spelling. For example, some applicants entered wrong spelling for their undergraduate university and we made corresponding correction.

After completing all the data pre-processing work, we obtained 45 attributes for each applicant (See Table 1) and we stored all these data into the MySQL database.
<table>
<thead>
<tr>
<th>1. idnum</th>
<th>2. reference_no</th>
<th>3. Year</th>
<th>4. name</th>
<th>5. ad_round</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. ad_result</td>
<td>7. ad_supervisor</td>
<td>8. ad_group</td>
<td>9. ad_degree</td>
<td>10. ad_hkpf</td>
</tr>
<tr>
<td>16. QS_on_ug</td>
<td>17. major_ug</td>
<td>18. major_ug_other</td>
<td>19. gpa_ug</td>
<td>20. gpa_ug_scale</td>
</tr>
<tr>
<td>21. rank_ug</td>
<td>22. university_pg</td>
<td>23. QS_pg</td>
<td>24. QS_on_pg</td>
<td>25. major_pg</td>
</tr>
<tr>
<td>26. major_pg_other</td>
<td>27. gpa_pg</td>
<td>28. gpa_pg_scale</td>
<td>29. rank_pg</td>
<td>30. interest1</td>
</tr>
<tr>
<td>31. interest2</td>
<td>32. interest3</td>
<td>33. english_tests</td>
<td>34. papers</td>
<td>35. hc</td>
</tr>
<tr>
<td>36. tc</td>
<td>37. toc</td>
<td>38. status</td>
<td>39. toefl</td>
<td>40. CET6</td>
</tr>
<tr>
<td>41. shortlisted</td>
<td>42. norm_gpa_ug</td>
<td>43. norm_gpa_pg</td>
<td>44. QSRanking</td>
<td>45. onQSRanking</td>
</tr>
</tbody>
</table>

Table 1: Attributes of applicant data

**Visualization**

HKUEDM includes following types of visualizations.
**Dashboard Visualization**

Dashboard Visualizations provides an overview on the application data of all applicants or a specific year of applicants chosen by the user. It uses different forms of charts to best present the information of the following three aspects:

1. Overview on the application trend of all applicants by the program type or the interest group they apply for.

![Dashboard Visualization](image)

**Figure 4: Application trend visualization 1**

![Applicant](image)

**Figure 5: Application trend visualization 2**

From the above two figures, the trend of all application in 2015 is displayed to users. The Figure 4 categorizes the applications and present the corresponding statistics of each group while the Figure 5 presents the number of applicants with
different interest group as their primary preferences. The pie chart clearly shows the difference between the size of applicants applying for each interest group. Each visualization also provides an entry point for users to access the list of applicants in that category.

2. Overview of the performance of all applicants based on their undergraduate GPA, and number of papers published.

![Applicants Chart](image)

Figure 6: Applicant Bubble Chart

Figure 6 is a bubble chart that plots each applicant as a point in a 2 dimensional coordinate. Three attributes of application data, including applicant’s undergraduate grade point average, number of papers published and his or her admission result, are shown in the graph. Detailed information of the applicant can
be seen when user mouse-over each point in the bubble chart. By clicking a specific point, users can visit the corresponding applicant’s application page to learn more about the applicant.

Applicant Detail Visualization

![Applicant Detail Visualization](image)

Figure 7: Applicant Detail View

This page presents detailed application data of a specific applicant. Other than showing the value for each attribute of the applicant, HKUEDM also presents the percentile of this value among all applicants of the same year.
Customized Visualization

HKUEDM also allows users to create customized visualization by using the bubble graph. Users can filter the dataset to use the application of a specific year or a specific interest group. They can also choose the attributes for the following parameters that plots each applicant as a bubble into the chart:

- X-axis: Undergraduate GPA, QS Ranking, Papers Published, TOEFL score
- Y-axis: Undergraduate GPA, QS Ranking, Papers Published, TOEFL score
- Z-axis: Gender, Research Interest 1, Admission Result, Year, Application Program Type, Admitted Program Type, HKPF

The X-axis and Y-axis is used to present numerical values while the Z-axis is used for presenting categorical value. For instance, Figure 8 includes all applicants with DB (Database) as their first research interest in the bubble chart. It can be seen from the chart that most applicants did not publish any papers. For the admitted applicants, they are sparsely distributed along the Y-axis while centering in the right of X-axis.
HKUEDM provides the two analysis methods, Decision Tree and Association Rule. For each method, HKUEDM can automatically do the analysis and present the results to users or allow users to do customized analysis by selecting specific dataset and setting different parameters.

**Decision Tree**

As mentioned in RELATED STUDIES, being a non-parametric supervised learning method that can be used for classification and regression, Decision Tree has become one of the most widely used data mining algorithm in the field of EDM. Decision Tree (the DT) constantly generates decision rules inferred from the data features and uses these rules to builds a model
to predict the value of a target variable, which can be the admission result of an applicant. The following features of Decision Tree algorithm make it suitable for HKUEDM:

- It can be applied on both numerical and categorical data.
- It is able to generate multi-dimensional output, and thus handle the multi-output problems.
- Outcomes can be visualized in a tree structure and can be easily interpreted.

The DT is implemented with the `DecisionTreeClassifier` class in scikit-learn, which uses an optimized version of CART (Classification and Regression Tree). CART constructs a binary tree using the attribute and threshold that yields the largest information gain at each node, where Information gain is defined as the mutual information $I(X; A)$ of $X$ and $A$ – i.e. the reduction in the entropy of $X$ (denoted as $H(X)$) achieved by learning the state of the random variable $A$, i.e. $IG(X,a) = H(T) - H(T|a)$. This suggest that the larger the information gain of an attribute is, the more important narrowing down the state of $X$ is.

Other than the default analysis, which generates a model that includes all the attributes and all the applicants in the database, the DT also allows users to do the following customizations.

- Dataset selection. Users can select the year of applicants to be included.
- Attributes selection. The potential attributes for selection are:
  - The Undergraduate Major: CS/Non-CS
  - Normalized Undergraduate GPA
  - Number of papers published before
  - TOEFL Score
  - QS Ranking of Undergraduate University
- Target class for classification selection. The potential classes for classification are:
• Admission result: Not Admitted / admitted to MPhil / admitted to PhD

• HKPF: No HKPF / HKPF Awardee

- Select parameter: Users can specify the maximum depth of the tree so as to avoid over-complex trees with unnecessarily great depth.

Figure 9 shows the interface of the DT. The DT uses a tree graph to visualize the decision rules with each node as a splitting criterion except for the leaf nodes, which represent the set of applicants that satisfy the rule along the path. Users can also mouse over to see the distribution over different classes. Apart from the dividing values of each attribute, the DT also generates a sorted list of attribute importance to help users compare the importance of each attribute used to classifier applicants into a specific class.

![Decision Tree Visualization](image)

**Figure 9: Decision Tree Visualization**

**Association Rule**

The Association Rule functionality (the AR) applies the Apriori algorithm, which is used for frequent item set mining and association rule learning over transactional databases, to
generate association rules which highlight significant trends underlying the data. In this project, we consider each attribute of an applicant as an item and each applicant as a transaction. By applying the Apriori algorithm, a "bottom up" approach is taken, where a set of attributes (items) form the item set and frequent subsets are extended one item at a time. The Apriori algorithm terminates when it cannot find further successful extensions. To count the candidate item sets more efficiently, breadth-first search and a Hash tree structure are used in the implementation. In this way, the Apriori algorithm proceeds by identifying the common applicant’s characteristics (frequent individual items) in the database and extending them to larger and larger item sets by adding applicant’s characteristics as long as those characteristics appear sufficiently often in the database. From each of these extensions, we could derive an association rule with relevant statistics for our analysis [7].

The AR first allows users to select a list of attributes to be used as items (See Figure 10). Since the Apriori algorithm can only be applied on binary attributes, the AR first transforms the numerical attributes into binary attributes by user’s selection of dividing value and uses it to separate the attribute into two categories. By default, the threshold value for each numerical attribute is the result obtained from the Decision Tree. For example, by default, the result of Decision Tree shows that 0.86 is the best value for normalized undergraduate grade point average that separate the applicants. Then the AR uses 0.86 to transform the normalized grade point average variable “norm_gpa_ug” into two items “norm_gpa_ug < 0.86” and “norm_gpa_ug >= 0.86”. In this way, each applicant (transaction) only has one of these two items.
After submitting the customized user input, the AR will present a few significant association rules (See Figure 11). For example, the item set with more than two items and highest support value will be presented along with the highest confidence rule within this item set. Meanwhile, the rules with being admitted (ad) as consequent item in a rule will also be presented so that users can clearly see which group of attributes influences the admission result most. The computation of support and confidence is based on the following information.
- **Support**
  
  - $\text{supp}(X)/N$
  
  - The proportion of data in the database which contains the item-set $X$.
  
  - An indication of how frequently an item-set appears in the database.

- **Confidence**
  
  - $\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X)$
  
  - The proportion of data that contains $X$ which also contains $Y$.
  
  - An indication of how often the rule has been found to be true.

The AR also provides users with a list of ranked item sets that have the greatest *support* value and more than two items (See Figure 12). From this table, users can easily identify what are the most common characteristics (frequent items) shared by the applicants. For example, we can see from the No.1 item set that 63% of the applicants have published fewer than 4 papers, come from an undergraduate university with a QS ranking higher than or equal to 112 and are rejected in the application.

![Ranked support itemset](image)

**Figure 12:** Association Rule ranked item sets by support
**Prediction**

**Logistic Regression**

This Logistic Regression functionality (the LR) uses the logistic regression, which is a regression model where the dependent variable is categorical, to build a classifier so as to classify applicants into different categories. The categorical dependent variable can be binary or multiclass, which corresponds to binary classification or multiclass classification (multinomial classification), respectively. The reason to implement the logistic regression is that the logistic regression model can easily help us measure the relationship between the categorical dependent variable, which could be the admission results or scholarship results in our project, and one or more independent variables, like the attributes of each applicant, by estimating probabilities. Moreover, by using the generated probabilities, we can rank the applicants, classify them into different results, and accordingly, make recommendations based on the classifications.

We implemented the Logistic Regression algorithm with the existing packages in *Scikit-learn*, the `sklearn.linear_model.LogisticRegression` Class, which uses the ‘liblinear’ library. For the multiclass classification, the algorithm uses the *one-vs-rest* (*OvR*) scheme. The specific results provided by this algorithm are incorporated in the following discussion of the detailed use case in the LR [8].
Figure 13: Customized User Input in the Logistic Regression

The LR first allows users to select a list of independent variables and the dependent variables to build the model (See Figure 13). In addition, users can specify the year of applicant data to be used as samples to build the regression model (Training model year) and the year of applicant to be used as test set to make predictions (Target year).

Figure 14: Logistic Regression decision function
After submitting the customized user input, the LR will first preprocess the applicant data to transform categorical variables to dummy variables. For example, the application program type by the applicant has three possible values, ‘phd’, ‘mphil’ and ‘either’. Accordingly, we use two dummy variables, ‘apply_phd’ and ‘apply_mph’ to represent this information. Then, the LR will generate the regression model based on the applicant data of the training year and present the results in a table consisting of the logistic regression decision function for each target class, the coefficient for each selected independent variable and the intercept. If it is a binary dependent variable, only the logistic regression decision function for one of the target class will be presented (See Figure 14). From this table result, users can easily find out whether the correlation between the dependent variable and each independent variable is positive or negative and which dummy variable is playing a more important role in the model.

If the dependent variable is the admission result, the LR will also display a table of applicant data, which shows the probability that the applicant is classified to be admitted, the value of each attribute of the applicant, the rank and the predicted result of the applicant based on its probability (See Figure 15). The probability of each class is computed using the logistic function. The rank is ranked from the highest probability to the lowest probability. To obtain the predicted result for each applicant, we first use the previous year data to find out the percentage of applicants being admitted. Then we use this percentage and the total applicants in the target year to obtain the estimated number of admitted applicants (n) for the target year. In the end, we predict the applicants with the highest n rank to be admitted. Moreover, the table allows users to sort the applicant data by any column so that users can easily inspect the data and look for potential insights. Users can also click into each applicant to see his or her detailed profile through a link.
RESULTS AND IMPLICATION

Decision Tree

The following two results obtained from the DT require our attention.

Result 1: Although the admission criteria for MPhil and PhD are similar, in general, admitted MPhil students have published less papers but they come from better universities, i.e. universities with higher QS ranking.

By analyzing each node in Figure 16, we obtained the following results:
- Node 5 contains the greatest number for both MPhil and PhD applicants and it represents the following decision rule:
  - Normalized Undergraduate GPA is in the range of [0.86, 0.98].
  - Number of papers published is smaller or equal to 4.
- The difference between Node 7 and Node 8 indicate that given the same GPA, MPhil applicants usually come from a university with higher QS Ranking.
- Node 6 shows that given the similar range of GPA, applicants with more papers published are more likely to be admitted to a PhD program.

Figure 16: Decision Tree on Admission

**Result 2:** HKPF Awardees over the previous years have distinctively higher GPA than other admitted students.
The Figure 17 displays the analysis on how important the Undergraduate GPA is in the determination of whether an admitted student can get HKPF. It shows that the almost all the HKPF Awardees have a higher than 0.9 normalized Undergraduate GPA.

![Decision Tree on HKPF](image)

**Figure 17: Decision Tree on HKPF**

**Visualization**

We found the following two results through the visualization of the applicant data.

**Result 1:** Although female applicants have an overall better performance in terms of GPA and Papers Published, more male applicants were admitted.
Figure 18: Visualization on the admitted students

The Donut Chart in Figure 18 shows that in the previous three years, more male applicants were admitted to HKUCS. However, as can be seen from the bubble chart, female applicants have an overall better performance in terms of their undergraduate GPA and Papers Published.

**Result 2:** Students admitted in earlier rounds generally have better performance.

Figure 19 shows that the relationship between the performance of admitted students in the previous three years and their admitted round. It can be seen that students admitted in the early first round generally have better performance in terms of Undergraduate GPA and Papers Published than students admitted later.
Association Rule

When we use the admission result, grade point average, QS ranking, number of papers published, research interest and applied program type with the threshold value obtained from Decision Tree as items and apply the Apriori algorithm on the applicant data of 2015 and 2016, we obtained a few rules that have implications.

\[
\text{‘normalized UG GPA > 0.86’, ‘published papers < 4’} \Rightarrow \text{admitted}
\]

Support = 0.10
Confidence = 0.15

Figure 19: Visualization on the admitted students 2

Figure 20: Association rule 1
The association rule 1 with a support value 0.10 shows that there are 10 percent of applicants who had normalized undergraduate grade point average greater than 0.86, published fewer than 4 papers and were admitted. Given that only 11 percent of applicants were admitted in 2015 and 2016, this item set almost represent the admitted students. Such a result indicates that a greater than 0.86 normalized undergraduate grade point average and fewer than 4 papers published can be used as a precondition to filter applicants for admission. Furthermore, since the confidence value is only 0.15, which means only 15 percent of the applicants who satisfied this precondition were admitted, such a precondition is only the first step filter and we need more filters to accurately predict the admission results. Such a result indicates that admission committee seems to give lower priority to applicants with more than three papers published. To some extent, this is reasonable since many applicants just graduated from their undergraduate education and they generally do not have chance to publish many papers. However, this does not reflect that they are not capable of doing in-depth research and their academic potential should be judged by other aspects. Moreover, other than the number of papers published, the quality of the papers should also be considered. For young applicants with limited experience in research, it is reasonable to suspect that the applicants with more papers published tend to have lower quality of those papers.
Figure 21: Association rule 2

The association rule 2 with a high confidence value 0.95 indicates that 95% of the applicants who had their normalized undergraduate grade point average less than 0.86, applied for PhD and their undergraduate university has a QS ranking greater than 111, were rejected. Such a rule can be used as a precondition to filter the applicants who are applying for PhD and applicants with lower grade point average and nor coming from a prestigious university should be given lower priority in terms of making admission decision.

**Logistic Regression**

When we use the applicant data of Year 2015 as the training data and the list of attributes show in the below figure for building the logistic regression classifier, we obtained a regression model (See Figure 22), which is the decision function for the target class of dependent variable, reject ($rej$).

![Logistic Regression decision function](image)

Figure 22: Logistic Regression decision function

\[ rej_i = 0.52 + 0.342apply\_phdi + 1.49apply\_mphi + 0.001qs\_ugi - 0.299major\_cs_i + 0.737attend\_pg_i + 0.036papers_i - 0.1norm\_gpa\_ugi + \epsilon_i \]

Since this is the decision function for the target class, reject, the higher the value of the function is, the more likely the applicant will be rejected. The analyses of each estimated coefficient are listed below.
• *apply_phd* and *apply_mph* both have positive estimated coefficient. However, since the estimated coefficient of *apply_phd* (0.342) is smaller than the one of *apply_mph* (1.49), it is more likely for an applicant applying for a MPhil program to be rejected than an applicant applying for PhD.

• *qs_ug* has a positive estimated coefficient. Such a result shows that the larger QS ranking of the applicant’s undergraduate university, the more likely he or she would be rejected, which is consistent with our expectation.

• *major_cs* has a negative estimated coefficient, which indicates that students who study Computer Science during their undergraduate are less likely to be rejected.

• *attend_pg* has a positive estimated coefficient, which reveals that those applicants who have completed a graduate study before receive no benefit in terms of their applications. Instead, they become more likely to be rejected. Such a result is surprising.

• *papers* has a positive estimated coefficient. Such a result is surprising since the more papers the applicants published, the more likely he or she will be rejected.

• *norm_gpa_ug* has a negative estimated coefficient, which shows that the applicants with higher grade point average are less likely to be rejected, which is also consistent with our expectation.

Moreover, we can extract similar information from the recommended applicants for admission provided by the Logistic Regression. We can see from the figure that the top 10 recommended applicants are all applying for PhD only instead of MPhil; their undergraduate universities all rank relatively high, which is equivalent to relatively small ranking index; they all major in Computer Science and they never publish paper before.
Therefore, based on the results provided by the Logistic Regression, we suggest that applicants majoring in Computer Science and only applying for PhD should be paid more attention to. Furthermore, the number of paper published is actually not a significant attribute to determine whether to admit an applicant and more paper published is not necessarily an advantage for the applicants.

**DIFFICULTIES AND PROPOSED SOLUTIONS**

One of the difficulties we encountered during the implementation of Decision Tree algorithm is overfitting. The algorithm was initially applied to the whole training data set. However, the decision tree grew too big with too many attributes considered and thus failed to provide useful information as expected. To solve the problem, we used 10 folds cross-validation, which is a method available in *Scikit-learn*. The data was divided into 10 sets randomly and the algorithms ran for 10 iterations. In each iteration we used 9 sets to train the data and 1 set to test the data. In this manner, the model was trained in multiple iterations with smaller data sets. Thus, the depth of the decision tree was limited and the accuracy increased.

Another potential difficulty is the paucity of the data available. Although we have collected the admission data of 2014, 2015, 2016 and 2017, the amount of data is still limited, which

<table>
<thead>
<tr>
<th>ID</th>
<th>Predicted Result</th>
<th>Rank</th>
<th>Actual Probability</th>
<th>apply_phd</th>
<th>apply_mph</th>
<th>qs ogs</th>
<th>major_cs</th>
<th>attend_gy</th>
<th>papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y2016</td>
<td>ad</td>
<td>1</td>
<td>0.381</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2017</td>
<td>ad</td>
<td>2</td>
<td>0.371</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2018</td>
<td>ad</td>
<td>3</td>
<td>0.371</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2019</td>
<td>ad</td>
<td>4</td>
<td>0.371</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2020</td>
<td>ad</td>
<td>5</td>
<td>0.371</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2021</td>
<td>ad</td>
<td>6</td>
<td>0.371</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2022</td>
<td>ad</td>
<td>7</td>
<td>0.371</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2023</td>
<td>ad</td>
<td>8</td>
<td>0.389</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2024</td>
<td>ad</td>
<td>9</td>
<td>0.389</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Y2025</td>
<td>ad</td>
<td>10</td>
<td>0.368</td>
<td>phd</td>
<td>-</td>
<td>15</td>
<td>CS</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 23: Logistic Regression recommended applicants
might create limitation in the model we build. To address the problem, we tried to make the system scalable enough to take in new admission data each year such that the models can be easily adjusted to be more robust.

**CONCLUSION**

This project aims at developing an educational data mining web application for the administrators and teachers of HKUCS to analyze the graduate student data so as to facilitate the admission decision-making process and answer questions that can be crucial to the admission strategies of HKUCS. An Educational Data Mining web tool (HKUEDM) has been developed and provides a set of functionalities, including data visualization, data analysis and outcome prediction. In addition, various analyses have been conducted on the application data and based on the analyses results, a few implications have been drawn. We believe this application will have a real impact on the HKUCS Admission Programme in the future and can be later adapted to help universities to admit undergraduate students, or help companies recruit new employees.
**CONTRIBUTION**

The detailed contribution of each group member to the project is listed in the following table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xu Fangyuan</td>
<td>Data visualization</td>
</tr>
<tr>
<td></td>
<td>Decision Tree Classification</td>
</tr>
<tr>
<td></td>
<td>Decision Tree Regression</td>
</tr>
<tr>
<td>Wu You</td>
<td>Database Management</td>
</tr>
<tr>
<td></td>
<td>Association Rule</td>
</tr>
<tr>
<td></td>
<td>Logistic Regression</td>
</tr>
</tbody>
</table>

Table 2: Contribution of each group member
REFERENCE


