COMP4801 Final Year Project

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

FINAL REPORT

Supervisor: Dr. Reynold Cheng

Written by: Xu Fangyuan (3035085530)

Group Member:

Wu You (3035086120)

Xu Fangyuan (3035085530)

Submission Date: 16th April 2017
DISCLAIMER

Information published on the web application of this project is used for educational purposes only. The Computer Science Department of the University of Hong Kong and the project team is committed to ensuring that all personal data involved in the project are treated with respect and safeguarded to ensure privacy. All personal identifiers have been removed from the website and the report.
ABSTRACT

Educational Data Mining (EDM) is an emerging field that uses statistical and machine learning algorithms to find patterns in massive educational data sets. Common problems targeted includes admission outcome prediction and student performance analysis. To ameliorate the admission process of the graduate programs in the Computer Science Department of the University of Hong Kong (HKUCS), this project applies educational data mining techniques to the admission data in the past four years. The objective 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. Based on the admission patterns that are found, the web application suggests a ranked list of candidates for admission decisions. The project also includes an experiment conducted on the past admission data with the functions provided by the web tool.
ACKNOWLEDGEMENT

The progress of the project and the final report would not have been possible without the kind support and help of different individuals and organizations. We would like to extend our sincere thanks to all of them.

We would like to express our gratitude to Dr. Reynold Cheng, our supervisor, for providing useful guidance for the direction of our project and timely support throughout the project.

We would like to express our thanks to Mr. Jiafeng Hu, who is highly involved in the admission process of the graduate programs in HKUCS, for attending an interview with us and providing us with his in-depth observations on the problem that we are trying to solve.

We would also like to express our appreciation for Mr. You Dongguang, a member of the project group of A Mobile and Intelligent Student Interview System for HKUCS, for providing with us the source code of last year’s project for study.
TABLE OF CONTENT

INTRODUCTION .................................................................................................................. 9

REQUIREMENT ANALYSIS ............................................................................................... 11

RELATED STUDIES ........................................................................................................... 13

SCOPE ............................................................................................................................... 14
  Extraction Layer ............................................................................................................ 14
  Analysis Layer ............................................................................................................... 15
  Prediction Layer ............................................................................................................. 16

DELIVERABLE ................................................................................................................... 17

CONTRIBUTION ............................................................................................................... 17

METHODOLOGY ............................................................................................................... 19
  Web Framework and Database ...................................................................................... 19
  Visualization .................................................................................................................. 20
  Data Mining .................................................................................................................. 20

SYSTEM STRUCTURE ...................................................................................................... 22
  Functional Framework ................................................................................................. 22
  Database Design .......................................................................................................... 23

FUNCTIONALITIES AND IMPLEMENTATION ................................................................... 24
  Extraction ....................................................................................................................... 24
    Data Collection .......................................................................................................... 24
    Data Pre-processing .................................................................................................... 25
  Visualization .................................................................................................................. 26
    Dashboard Visualizations ........................................................................................... 26
    Applicant Detail Visualization ..................................................................................... 29
    Customized Visualization ........................................................................................... 29
  Analysis .......................................................................................................................... 31
    Decision Tree .............................................................................................................. 31
    Association Rule ......................................................................................................... 34
  Prediction ....................................................................................................................... 37
    Logistic Regression ..................................................................................................... 38
EXPERIEMNT AND RESULT..........................................................41

Decision Tree Classification..........................................................41

1. The admission criteria for MPhil and PhD are similar. However, generally MPhil students have published less papers but they come from better universities. ..........................................................41

2. HKPF Awardees over the past years have distinctively higher GPA than other admitted students. ..........................................................42

Visualization.................................................................................43

1. More male applicants were admitted while female applicants have an overall better performance in terms of GPA and Papers Published. ............43

2. Students admitted in earlier rounds generally have better performance. Moreover, a deficiency in one field might be compensated by satisfying performance in another..........................................................44

Association Rule..........................................................................45

1. There are certain similarities in terms of the academic performance among the admitted applicants. However, factors other than the attributes in application data also affect the admission decision. ......................46

2. The number of papers published plays a relatively insignificant role in the admission decision..........................................................46

3. For PhD applicants, there tends to be threshold values for the Undergraduate GPA and QS Ranking for undergraduate university..............47

Logistic Regression.........................................................................47

1. Students majoring in Computer Science and only applying for PhD should be paid more attention to. ..........................................................47

CONCLUSION..................................................................................49

REFERENCE...................................................................................50
LIST OF FIGURES

Figure 1. The table presenting applicant information in the current system .......... 12
Figure 2. User interface of HKUCS EDM tool ................................................. 17
Figure 3. The functional framework of the system ............................................. 22
Figure 4. Class Diagram of Database ............................................................ 23
Figure 5. Application trend visualization 1 ....................................................... 27
Figure 6. Application trend visualization 2 ....................................................... 27
Figure 7. Applicant Bubble Chart ................................................................. 28
Figure 8. Applicant Detail Visualization ......................................................... 29
Figure 9. Customized Bubble Chart .............................................................. 30
Figure 10. Decision Tree Classification Interface ............................................ 34
Figure 11. User Customization for Association Rule ....................................... 36
Figure 12. Association Results ................................................................. 37
Figure 13. Ranked Association Rules ............................................................ 37
Figure 14. User customization in Logistic Regression ...................................... 40
Figure 15. Logistic Regression Decision Function .......................................... 40
Figure 16. Logistic Regression Applicant Table ............................................ 41
Figure 17. Decision Tree on Admission Result ............................................ 42
Figure 18. Decision Tree on HKPF Result ...................................................... 43
Figure 19. Visualization on the Admitted Students ......................................... 44
Figure 20. Visualization on the Admitted Students 2 ..................................... 45
Figure 21. Association Rule 1 ................................................................. 46
Figure 22. Association Rule 2 ................................................................. 47
LIST OF TABLES

Table 1 Project Contribution .................................................................................................................. 18
Table 2 Attributes of Applicant Data ..................................................................................................... 26
GLOSSARY

**EDM:** Educational Data Mining

**HKUCS:** The Computer Science Department of the University of Hong Kong

**HKUEDM:** Educational Data Mining Tool of the University of Hong Kong (i.e. the name of the web tool developed in the project)

**HKPF:** Hong Kong PhD Fellowship
INTRODUCTION

Educational data mining (EDM) is a growing research area that aims at studying educational problems by exploring data originating from educational context. [1]. It is generally acknowledged that substantial valuable information 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). Every year, HKUCS has to scan through a large number of application to the graduate programs and select a small number of suitable candidates. For instance, in the year of 2016, only 8 out of 178 applicants are given an offer. Given the complexity of the large range of factors considered, it is not an easy task to evaluate a candidate. Around 30 parameters of different types, such as their GPA, their English skills and their research interests, correlate with each other and all contribute to the decision making.

To facilitate the admission process, there exists a web system that allows professors to view application details, make comments and maintain application status. However, the existing admission system doesn’t fully utilize the application data as a scientific support to improve the efficiency of the admission decisions. Professors have to look at each attribute of the application themselves to evaluate the candidates, which is rather time consuming. Apart from that, there is a lack of intuitive tools that present the underlying patterns of application and admission trend over all applications, which will be useful for the decision making.

Therefore, this projects aims to develop an EDM web application (HKUEDM), which can make use of the graduate students’ data to conduct data extraction, visualization and analysis. Based on the analysis, the application ranks applicants by their performance and predicts admission results. With these functionalities, HKUEDM application finds important patterns and trends to provide insights into the admission strategies as well as to facilitate the admission process of HKUCS.
Moreover, the project also includes a data mining analysis conducted on the existing application data, which contains applicants from the past 4 years. Insights of the application data and admission pattern as well as trends across multiple years have been discovered by data mining algorithms as well as data visualizations implemented in the HKUEDM.

The rest of the report proceeds as follows. It first discusses the requirement analysis conducted, which includes an overview on the existing online system. Then it examines the previous studies in the field of Educational Data Mining and their relevance to this project. Following that it provides an overview on the scope of the project and the deliverables. Then it discusses the methodology and technology used in the project. To provide a clearer picture on the implementation, it then illustrates the system structure and talks about the implementation of each function in detail.

As for the analysis part, it presents the result of the data mining analysis on the application data in the past 4 years. Based on the result, it discusses the implication on application and admission pattern. To conclude, it provides possible directions for future work.
To better understand the crucial problems concerning the admission process, we gathered requirements from the potential users of the system. We conducted interviews with Dr. Reynold Cheng, the head of graduate admission in HKUCS, and Mr. Jiafeng Hu, who is responsible for reviewing applications. 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 will be very helpful if the web application can extract useful information and make sensible recommendations to them.

Apart from that, we also assessed the usability of the existing system. First of all, we found it rather difficult to navigate through a long list of applicants, which makes it troublesome to quickly identify a suitable candidate.

Moreover, we found it also difficult to evaluate an applicant given all the attributes presented in a paper-like format (See Figure 1). Apart from that, we found that the existing graphs in the system only contains comparisons based on one attribute. However, since many of the attributes are inter-correlated and they all contribute to the result together, this kind of graphical presentation fails to provide a comprehensive representation of the data. Thus, we figured out that data visualization would be a useful way to increase efficiency in the admission process.

Based on the requirements captured and the problems found, we derived the functions mentioned in **SCOPE** and pick out the following questions:

1) Whether an applicant will be admitted or not;
2) Whether an applicant will receive Hong Kong Postgraduate Fellowship or not;
3) Whether an applicant will be admitted to a specific professor or a research interest group or not.

![Student Applicant info table]

Figure 1. The table presenting applicant information in the current system
RELATED STUDIES

In general, the techniques most commonly used in Educational Data Mining include Association Rule, Classifications and Regressions [1]. There are some existing projects that build data mining tools with these techniques and generate satisfying outputs. They are listed as follows with an evaluation of their relevance to this project.

Mashat et al. demonstrated the use of Association Rule in EDM. They applied the Apriori algorithm, one of the association rule mining algorithms, on the university admission to extract relations between different attributes like GPA and courses taken by the students [2]. These relations provide insights to compare the importance of different assessment criteria during the admission process. In the EDM tool, we implemented an Association Rule that compares various attributes of an application.

Feng et al. established a university admissions decision-making model by utilizing the Self Organizing Map (SOM) neural network, cluster analysis, Association Rule and Fayyad data mining model [3]. Their model astutely takes geographical data into consideration and develops a new reference for admission scheme and propaganda. However, given the limitation on the data size, which contains less than 1000 applicants across 4 years, we didn’t implement the SOM neural network.

Fong et al. designed a hybrid web recommender system of Neural Network and Decision Tree Classifier to predict university admission in Macau [4]. They applied back propagation algorithm to train and build a learning model that calculates a weighted importance of different attributes. Then they run 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 built stops to decrease. Referring to their model, we have implemented a Decision Tree classifier that includes 5 attributes and 2 classifications on the application data.
SCOPE

Given the complexity of the problem, there are different perspectives to understand the data. Thus, the scope of the project is divided into three layers.

Extraction Layer

The first layer is to extract data that are identified as useful based on users’ requirements and present them in an understandable way. The visualization mainly contains two parts: visualization on a set of applicants and visualization on a specific applicant.

1) **Dashboard Visualizations** - Visualization on a set of applicants provides an overview of the trend and pattern.
   It includes different types of charts, such as bubble chart, pie chart and line chart. Unlike the existing charts in the system which focus on one attribute, HKUEDM generates charts that concern multiple correlated attributes, making it visually easier to compare and evaluate an applicant. Apart from that, the visualization also helps the user to navigate through the complex data set by providing interactive functions. For example, users can mouse-over a specific applicant to see further information or even click to be re-directed to a detail applicant page without the trouble of locating him/her in a lengthy applicant list.

2) **Applicant Detail Visualization**
   The visualization on a specific applicant highlights the important information from a lengthy application and helps the user to evaluate the applicant.

3) **Customized Visualization**
   As the data set includes a rather wide range of different attributes, there are many possible ways to analyse them and create visualizations accordingly. To facilitate the analysis, the tool provides customized visualizations. Customizations can be made in the following aspects:
Mining HKUCS Graduate Student Data: Extraction, Analysis, Prediction

i) The set of applicants to be visualized
ii) The attributes to be visualized
iii) Type of visualization

This function provides users with more flexibilities and creativities on the data analysis.

Analysis Layer

With the data extracted, HKUEDM provides users with different methods to make analysis and discover patterns in the application data. It implements a set of data mining algorithms. On top of that, it has a user-friendly interface for the users to easily apply the algorithms to conduct analysis and make customizations. The following algorithms are implemented:

- **Decision Tree.** This function runs the Decision Tree Algorithm on a set of data based on their value of a list of attributes. It builds a decision tree that classifies the data into different classes, which includes decision rules that shows the importance of attribute and the dividing values.

- **Association Rule.** This function considers attribute as items and applicant data as transaction. It finds the frequent itemset (a set of attributes) that appear often among all the application data based on statistical values such as support and confidence. Based on the itemset found, it presents association rules that reveal the relation between different attributes.

By default, HKUEDM runs the data mining algorithms and generates a model with the most satisfying result. The result is visualized in a simple and direct way such that users
without prior knowledge to data mining can also easily get some insight from the data with such a one-click approach.

Apart from that, to cater the different needs of potential users, it also allows user to generate their own models and conduct customized analysis. The tool provides an interface for them to specify a set of parameters such as the algorithm and threshold value to use (if any) and the attributes to include.

The analysis results, which include a lot of statistics, are summarized and visualized in different graphical forms such that it can be easily understood by the users and thus presented to a larger range of audience.

**Prediction Layer**

HKUEDM builds a recommendation system supported by Logistic Regression algorithm to predict applicants’ admission results. Based on the pattern found in Analysis Layer, the recommendation system processes the incoming application data and classifies the applicants into different categories. The functions provided are listed as follows:

- **Logistic Regression.** This function builds the recommendation system and processes the incoming application data with the system. It allows users to specify the data to be use. The recommendation result is presented as a ranked list of applicant based on probabilities predicted by the system and served to other layers of the web tool.

- **Outcome Prediction.** This function presents the results of Logistic Regression in the form of a ranked list of applicants to users as 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 tool (HKUEDM). The web tool communicates directly with a database and provides a user-friendly interface (See Figure 2) with a set of functionalities that help make sense of the data. It also produces an analysis on an experiment conducted with the application data in the year of 2014, 2015, 2016 and 2017 using HKUEMD. The experiment result and discussion can be found in EXPERIMENT AND RESULT.

Information about the project can also be found on the project website: http://i.cs.hku.hk/fyp/2016/fyp16019/

CONTRIBUTION
The detailed contribution of each group member to the project is listed in the Table 1.

<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 1 Project Contribution
METHODOLOGY

This section introduces and discusses the choice of technology in the following three areas:

1) Web Framework and Database
2) Visualization
3) Data mining algorithms

Web Framework and Database

The web tool is implemented with Django as the web framework, following the model-view-template (MVT) architectural pattern. Django’s reusability and scalability of components are one of the major reasons for our choice, which allows us to build a complex, database-driven web application easily. Moreover, we can easily find solutions when encounter difficulties during development not only through the detailed documentation but also through the large and supportive Django community.

Data are stored in a local MySQL database. MySQL is a lightweight database that provides all features needed in the project. Moreover, it is supported by Django and thus reduces the trouble of integrating the database with other parts of the web tool.

1 https://www.djangoproject.com/start/overview/
Visualization

To create the data visualization, we chose d3.js, a JavaScript library that combines powerful visualization components and a data-driven approach to DOM (the Document Object Model for HTML).

As one of the most widely used JavaScript libraries for producing dynamic, interactive data visualization, d3.js provides a lot of choices on the types of visualizations and interactive functions. With a set of easy-to-use APIs, it is also low-level enough such that there are many flexibilities for users to customize the visualization with their own creativities.

Despite of the rather steep learning curve, there are many examples and resources available online for us to study and adapt. For example, we also used dimple, which is an object-oriented API powered by d3.js.

Other visualization libraries such as morris.js and chart.js are also used.

Data Mining

We handled the data mining algorithms with Python. 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 lower 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.

We have decided to use existing libraries to implement the data mining algorithms instead of writing them ourselves. The main reason is that this not only saves us a
considerable amount of time but also provides us with access to a larger range of algorithms that are more robust and powerful. Among the existing data mining libraries for Python, we chose *Scikit-learn* [7] for the following reasons.

First of all, it contains different data mining modules that cover the full scale of common data mining process, including data pre-processing, classification and regression. These functions allow us to implement the algorithms mentioned in RELATED STUDIES as well as conduct analysis such as model selection. Moreover, it is designed to interoperate with other numerical and scientific libraries as well as visualization libraries, which is also useful for this project. The libraries used along with scikit-learn in the project are listed as follows:

- **Numpy** [8] is an extension to the Python programming language, adding support for large, multidimensional arrays and matrices as well as high-level mathematical functions. It is used to deal with the large range of attributes in the admission data.

- **SciPy** [9] is an open source Python library used for scientific computing that contains modules for clustering, spatial data manipulation, optimization and etc. Apart from that, it also contains functions for splitting data-sets.

- **GraphViz** [10] is a package for drawing graphs. It is used to visualize the results generated in the data analysis in the project.

Last but not least, as a project started by Google, *Scikit-learn* not only provides powerful algorithms that are constantly maintained but also has a large user group and a lot of online resources.
SYSTEM STRUCTURE

Functional Framework

Figure 3 illustrates the framework of the functionalities of the web tool. There are mainly three layers: Extraction Layer, Analysis Layer and Prediction Layer. For each layer, the functions implemented are listed as well as their major sub-functions. The communication between layers and the database are also included.

Figure 3. The functional framework of the system
Database Design

Figure 4. Class Diagram of Database

Figure 4 illustrates the database design, primarily including the following model:

1) *applicants*: This is the model storing the application information. It contains 45 columns, each corresponding to an attribute of the applicant.
2) **auth_user**: This is the model for the user of the web tool. It relates to other models concerning the permission and authentication of a user.

**FUNCTIONALITIES AND IMPLEMENTATION**

**Extraction**

**Data Collection**

We have collected the application data for Postgraduate program in HKUCS from 2014 to 2017. The data contains the following information:

1) The field values in the application submitted by the applicant
2) The admission result and admission information of an applicant
3) The list of HKPF Awardees

Apart from that, we also collected the most up-to-date QS Work University Rankings by Subject 2016 – Computer Science & Information Systems. We then imported the ranking into the applicant’s model to rank the previous educational institutions mentioned in their application.

---

Data Pre-processing

To avoid misleading results caused by the quality of data, we first performed pre-processing on the raw data collected to handle the following problems: out-of-range values, missing values, inconsistent data scale, inconsistent data format, etc.

For numerical attributes of different scales, including GPA and QS Ranking for universities, we used standardization methods to convert them into comparable values. For categorical attributes, such as research interest and applied program type, we encoded them into digits.

To handle missing values of certain attributes, we discarded the noisy data when considering that specific attribute. For instance, we added an attribute named \textit{qs\_on\_ug} to indicate whether the undergraduate university of the applicant is listed on the QS Ranking.

We then handled the problem of inconsistent formats to extract useful information. For instance, an applicant’s English test results are stored as a string in the raw data. We then created a parse to extract the value for TOEFL, GRE, IELTS and CET6 and stored them in the database separately.

After pre-processing, there are in total 45 attributes stored in the database for each applicant.
Visualization

The web tool includes different types of visualizations. This section talks about their functionalities and implementations.

Dashboard Visualizations

Dashboard Visualizations provides an overview on the application data of all application data in the database or a specific year chosen by the user. With different forms of charts, it presents the information of the following aspects:

1. Overview on the application trend of all applicants: program type, interest group etc.
2. Overview of the performance of all applicants based on: GPA, Papers Published

**Application Trend**

![Figure 5. Application trend visualization 1](image1)

![Figure 6. Application trend visualization 2](image2)

For example, the above two figures illustrate the trend over all application in 2015. Figure 5 categorizes the applications and present the corresponding statistics of each group while the second one presents the number of applicants with different interest group as their primary preferences. Figure 6 illustrates clearly comparisons between the size of applicants from different groups.
Each gadget also provides an entry point for access the list of applicants in that category. For example, by clicking into the second gadget in Figure 5, a table of all applicants who applied for PhD in 2015 will be displayed.

**Applicant Performance**

![Applicant Bubble Chart](image)

*Figure 7. Applicant Bubble Chart*

The bubble chart plots all applicants as points in a 2D coordinate. Three dimension of application data is presented: Undergraduate GPA, Papers published and Application result. Application detail will be displayed when user mouse-over the bubble chart. By clicking a specific bubble, the page will be re-directed to the corresponding applicant’s detail application page.
Applicant Detail Visualization

<table>
<thead>
<tr>
<th><strong>Program:</strong> PhD</th>
<th><strong>Research Interest</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supervisor:</strong></td>
<td>Interest 1: PL</td>
</tr>
<tr>
<td><strong>Group:</strong> Algorithms</td>
<td>Interest 2: NI</td>
</tr>
<tr>
<td><strong>GPA:</strong> 3.6600 / 4</td>
<td>Interest 3: NI</td>
</tr>
<tr>
<td><strong>QS Ranking:</strong> 15</td>
<td></td>
</tr>
<tr>
<td><strong>90% among all applicants</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Papers published:</strong> 0</td>
<td></td>
</tr>
<tr>
<td><strong>33% among all</strong></td>
<td></td>
</tr>
<tr>
<td><strong>TOEFL:</strong> 90</td>
<td></td>
</tr>
<tr>
<td><strong>IELTS:</strong> N/A</td>
<td></td>
</tr>
</tbody>
</table>

**Teacher’s Comments**

@cs.hku.hk

Figure 8. Applicant Detail Visualization

This page presents detailed information of an applicant. It presents the percentile of his value for certain values among all applicants of the same year. For instance, this applicant has a GPA higher than 60% of all applicants in 2016.

Customized Visualization
The web tool also provides a customized visualization. Users can filter the dataset to see the application of specific year(s) or a specific interest group. They can also choose the values for the following parameters that plots applicants as bubbles onto a chart:

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

The first two axes present numerical values while the last one presents categorical value. For instance, Figure 9 presents the Application Program Type of all applicants with CF as the first research interest. It can be seen from the chart that there are more PhD applicants, especially for those who have published comparatively more papers.

![Customized Bubble Chart](image)

**Figure 9. Customized Bubble Chart**
Analysis

The web tool provides different types of analysis methods. For each method, the web tool automatically analyses the data and generates a model. It also allows users to customize on attribute value to generate different models. This section introduces the analysis methods, the functionalities and implementations.

Decision Tree

Algorithm overview

As mentioned in RELATED STUDIES, Decision Tree is one of the most widely used data mining algorithm in the field of EDM. It is a non-parametric supervised learning method used for classification and regression. It creates a model that predicts the value of a target variable (in this case, the admission result of an applicant) by learning simple decision rules inferred from data features. The following advantages make it suitable for the web tool:

- It is able to handle both numerical and categorical data.
- It is able to handle multi-output problems.
- Outcomes can be visualized and thus easily interpreted.

The Decision Tree Classification function in web tool is implemented with the DecisionTreeClassifier class in sklearn. It uses an optimized version of CART

---

3 Supervised learning is the machine learning task of inferring a function from labeled training data.
(Classification and Regression Tree), which constructs a binary tree using the feature and threshold that yields the largest information gain at each node.

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.

$$IG(X,a) = H(T) - H(T|a)$$

A large information gain of an attribute means that it is relatively important in narrowing down the state of $X$.

**Customization**

Users can filter the year(s) of applicants to be included as well as select from the following set of features:

- The Undergraduate Major: CS/Non-CS
- Normalized Undergraduate GPA
- Number of papers published before
- TOEFL Score
- QS Ranking of Undergraduate University

The following two classification can be made based on user selection:

- Admission result: Not Admitted/MPhil/PhD
- HKPF: No HKPF/HKPF Awardee

By default, it generates a model that includes all the features and all applicants in the database.

**Parameter Selection**
One of the disadvantages of decision trees is that they can create overly complex trees, i.e., those with unnecessarily great depth, which don’t generalize the data very well. To avoid this problem, the tool specifies a range of maximum depth: [3, 4, 5] and outputs the one with the highest score. The score is computed as subset accuracy, i.e., whether the predicted label for a sample matches its own label. It is implemented with the GridSearchCV class in sklearn.⁵

**Visualization**

Figure 10 is the interface of **Decision Tree Classification**. The Graph visualizes 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. By mouse over, the distribution over different classes is displayed.

Apart from the dividing values of attributes, the **Decision Tree Classification** also generates a sorted list of feature importance in making the decision.

Association Rule

Algorithm overview

The Association Rule function is implemented with an Apriori algorithm package\(^6\), which is commonly used for frequent itemset mining and association rule learning over transactional databases. The algorithm generates association rules with high support and confidence (frequent itemset), which reflects the underlying association between different attributes in the dataset. In this project, each dividing value of an application attribute is considered as an item (e.g. QS Ranking < 300) while each applicant as a transaction. In particular, successful admission (ad_result = “ad”) and recipient of HKPF (ad_hkpf =1) are considered consequent item.

The Apriori algorithm extends the frequent itemset by one more item at a time and terminates when no more successful extensions can be found. An extension is

---

\(^6\) See [https://pypi.python.org/pypi/apyori/1.1.1](https://pypi.python.org/pypi/apyori/1.1.1)
considered successful if the extended itemset appears sufficiently often in the database. To increase the efficiency of the algorithm, breadth-first search and a Hash tree structure are used.

**Item Encoding**

As the algorithm implemented in the web tool only handles binary attributes, numerical attributes are first transformed into binary ones, divided by a threshold value. By default, the threshold value for each numerical attribute is obtained from the model generated in Decision Tree. For example, the dividing value of “Undergraduate GPA <=0.86” illustrated in Figure is used as the threshold value to split the values of the attribute “norm_gpa_ug”.

**Customization**

Users can select a list of attributes to be included as items (See Figure 11). They can further specify the threshold value for each attribute. If not specified, the selected attributes are encoded in the way mentioned in **Item Encoding**.
The result generated by the Association Rule is presented in the form of significant association rules, i.e. itemset with more than two items and the highest support value. The highest confidence rule within this itemset is also presented (See Figure 12).

Apart from that, rules with “admitted” as a consequent item in a rule are also presented such that users can clearly see what are the attributes that influence the admission result the most.

The computation of support and confidence is based on the following information.

- **support**
  - supp(X)/N
  - It is proportion of transactions which contains the item-set X over all transactions.
○ This is an indication of how frequently an item-set appears in the database.

● confidence

○ \( \text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X) \)

○ It is the proportion of transactions that contains \( X \) which also contains \( Y \) over the transactions that contains \( X \). \( Y \) is the consequent item, e.g. “admitted”.

○ It is an indication of how often the rule has been found to be true.

Figure 12. Association Results

Apart from that, a list of ranked itemsets with more than two items is also presented (See Figure 13). The association reveals the common characteristics shared by a group of applicants (e.g. applicants who are admitted). For example, the first item set shows that 63% of all applicants have published fewer than 4 papers, come from an undergraduate university with QS Ranking higher than or equal to 112 and were rejected.

Figure 13. Ranked Association Rules

Prediction
Logistic Regression

Algorithm overview

The **Logistic Regression** functionality uses the logistic regression, a regression model, whose dependant variables are categorical. Depending on whether the classification is binary or multiclass (multinomial classification), the categorical dependent value is binary or multiclass accordingly. Logistic regression measures the relationship between the categorical dependent variable, i.e. admission results or scholarship recipient, and one or more independent variables, i.e. the attributes of each applicant. The model generated can give useful insight to the admission process. Moreover, the model generates a probability for each input (i.e. the attribute values of an applicant), which can be used to estimate the admission and scholarship result. The estimation can serve as a useful reference to the application reviewers.

Implementation

In **Logistic Regression**, Logistic regression is implemented with the `sklearn.linear_model.LogisticRegression` Class in [Scikit-learn](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html). For multiclass classification, the algorithm adapts the *one-vs-rest (OvR)* scheme.

Two set of applicant data are used in **Logistic Regression**:

1. The training year, which is used as sample to build the regression model
2. The target year, which is used as testing set to make predictions

---

**Logistic Regression** pre-processes both sets to transform categorical variables to dummy variables. For instance, the application program type has three possible values ‘phd’, ‘mphil’ and ‘either’, which are transformed into two dummy variables ‘apply_phd’ and ‘apply_mph’. It then generates the regression model based on the applicant data of the training model year. The model generated predicts the probability of admission for applicants in the target year.

**Logistic Regression** then calculates the percentage of applicants admitted in the training year, which is used to obtain the estimated number of admitted applicants($n$) in the target year. **Logistic Regression** presents the $n$ applicants with highest probabilities to be admitted.

**Logistic Regression** also provides an API that can communicate with the other layers of the web tool (e.g. Visualizations). The API provides a ranking list of recommended applicants based on the selected attribute and training year data selected by users.

**Customization**

Users can select a list of independent variables and the dependent variable to build the model. They can also specify the set of applicant data (e.g. applicants in 2016) as samples to build the regression model (Training model year) and the set of applicant data to be used as testing set to make predictions (Target year).
Result presentation

The results are presented in a table (see Figure 15), which consists

1) the logistic regression decision function for each target class
2) the coefficient for each selected independent variable, and
3) the intercept

Note that if the dependent variable is binary, the table only presents the logistic regression function for one of the target class.
If the dependent variable is the admission result, a table of ranked applicants (See Figure 16) will also be displayed, along with the following information:

1) The predicted result
2) The rank based on the probability that the applicant will be admitted
3) The probability that the applicant will be admitted
4) The value of the independent variables

Users can sort the applicant’s data by the value of each column in the table, which can be useful to see the overall trend of a specific attribute. The table also provides an entry for detailed application information of the applicants in the data.

**Figure 16. Logistic Regression Applicant Table**

**EXPERIEMENT AND RESULT**

**Decision Tree Classification**

1. The admission criteria for MPhil and PhD are similar. However, generally MPhil students have published less papers but they come from better universities.
In Figure 17, node 5 contains the greatest number for both MPhil and PhD applicants. It represents the following decision rule:

1) Normalized Undergraduate GPA is in [0.86, 0.98].
2) Number of papers published is smaller or equal to 4.

Comparing node 7 and node 8, it can be seen that given the same GPA, MPhil applicants are from universities with higher QS Rankings. Node 6 reveals that given the similar range of GPA, applicants with more papers published are more likely to be admitted to a PhD program.

2. HKPF Awardees over the past years have distinctively higher GPA than other admitted students.

Figure 18 shows the analysis of Undergraduate GPA on whether an admitted student can get HKPF. It shows that the majority of HKPF Awardees have normalized Undergraduate GPA higher than 0.9.
Visualization

1. More male applicants were admitted while female applicants have an overall better performance in terms of GPA and Papers Published.
Figure 19. Visualization on the Admitted Students

Figure 19 illustrates that in the past three years, more male applicants were admitted. However, as can be seen from the bubble chart, female applicants have an overall better performance in the two fields captured.

2. Students admitted in earlier rounds generally have better performance. Moreover, a deficiency in one field might be compensated by satisfying performance in another.

Figure 20 illustrates the relationship between the performance of admitted students in the past three years and their admitted round. It can be seen that students admitted in the early 1st round generally have better performance in terms of Undergraduate GPA and Papers Published. Moreover, even though an applicant, like Y16A02, has as
relatively low GPA, his performance in the other field is better than most of the other admitted students.

Figure 20. Visualization on the Admitted Students 2

Association Rule
1. There are certain similarities in terms of the academic performance among the admitted applicants. However, factors other than the attributes in application data also affect the admission decision.

![Association Rule 1](image)

**Figure 21. Association Rule 1**

Association rule 1 is generated by Apriori algorithm on the applicant data of 2015 and 2016 with the following set of attributes: Undergraduate GPA, QS Ranking, Number of Papers Published, Research Interest and Applied Program Type. The threshold value is obtained from Decision Tree.

With a support equal to 0.1, it indicates that 10 percent of all applicants satisfied the above rules and they were admitted, which is a significant proportion given that there are only 11 percent of applicants admitted in total. However, only 15 percent of the applicants who satisfied these rules were admitted. Thus, the above rule can be used to filter applicants in the earlier stage but cannot determine the admission result on its own. More evaluations are needed to determine whether the applicant should be admitted or not.

2. The number of papers published plays a relatively insignificant role in the admission decision.

Association Rule 1 also indicates that the number of papers published is not of great importance to the admission decision. It is rather reasonable given that most
applicants just got their Bachelor degree at the time of applying and didn’t have the change to publish papers in their undergraduate study. However, a low number of papers published doesn’t indicate that the student if not capable of doing research. It is also important to note that apart from the quantity, the quality of the papers published should also be considered.

The result of Logistic Regression also reveals that a relatively large number of papers published is not necessarily an advantage for the applicant. (See Logistic Regression)

3. For PhD applicants, there tends to be threshold values for the Undergraduate GPA and QS Ranking for undergraduate university.

\[
\text{\{'normalized UG GPA < 0.86', 'apply PhD', 'UG QS ranking > 112'\} } \rightarrow \text{\{reject\}} \\
\text{Support = 0.15, Confidence = 0.95}
\]

Figure 22. Association Rule 2

With a high confidence equal to 0.95, the rule indicates that most of the PhD applicants with normalized undergraduate GPA lower than 0.86 and attended a university with QS Ranking lower than 111, were rejected. Thus, the above rule can be used as a filtering criteria to select applicants for to be shortlisted in earlier rounds.

Logistic Regression

1. Students majoring in Computer Science and only applying for PhD should be paid more attention to.
The above logistic regression model is generated with the following input:

1) Training data: Applicants in 2015

2) List of independent variable: apply_phd, apply_mph, qs_ug, major_cs, attend_pg, papers, norm_gpa_ug

3) The dependent variable: admission rejected

The decision function is:

\[
\text{rej}_i = 0.52 + 0.342 \text{apply}_\text{phd}_i + 1.49 \text{apply}_\text{mph}_i + 0.001 \text{qs}_\text{ug}_i \\
- 0.299 \text{major}_\text{cs}_i + 0.737 \text{attend}_\text{pg}_i + 0.036 \text{papers}_i \\
- 0.1 \text{norm}_\text{gpa}_\text{ug}_i + \epsilon_i
\]

For each attribute, the higher the coefficient, the more likely the applicant will be rejected. Several insights can be obtained:

1) The estimated coefficient of apply_phd is significantly smaller than the one of apply_mph. Thus, an applicant applying for PhD is more likely to be admitted than one applying for MPhil.

2) The coefficient of major_cs is negative, which indicates that students who study Computer Science in their undergraduate study are more likely to be admitted.

3) In contrast, papers has a positive estimated coefficient, indicating that even if a relatively large number of papers published, the applicant might still be rejected. The result is consistent with the analysis in Association Rule.
Such insights are consistent with the information presented in the list of recommended applicants for admission. The top 10 recommended applicants share the same characteristics:

1) They are PhD applicants.
2) The QS Ranking of their undergraduate study is relatively high.
3) They major in Computer Science in undergraduate study.
4) They have never published any paper.

**CONCLUSION**

This project applies Educational Data Mining techniques to improve the admission process of the Postgraduate Program in HKUCS. An Educational Data Mining web tool (HKUEDM) has been developed and it provides different functions, including data visualization, data analysis and outcome prediction. Apart from that, experiments have
been conducted on the available application data. Based on the result, a list of insights has been discussed in the report as well.

However, given the limited time of the project, there remains room for improvement. The system can be further developed in the following aspects:

1) More Visualization options can be included, such as bar chart and area chart which shows comparisons between categories.

2) Currently, the visualization in HKUEDM can present at most three attributes in a graph. However, visualization that present the relationship between multiple attributes (e.g. more than 4) in a graph can also be useful for discovering their relationship. This requires more study and creativity to design the form of visualization.

3) More attributes can be used for data analysis. For instance, the feedback from the helper and the teachers can also be analysed and considered as a criterion for evaluation.

4) A function that improves the data mining model with new data can be included. As one of the major challenges facing the development is insufficiency of data, such a function can make the system more scalable and sustainable.

REFERENCE


