香港大學計算機科學系

THE UNIVERSITY OF HONG KONG

DEPARTMENT OF

COMPUTER SCIENCE

# COMP4801 Final Year Project

## HKUCS Graduate Admission Data Analysis:

## A Multimedia Mining Approach

### Final Report

3035124441 Wang, Michelle Yih-chyan

3035124829 Song Yi Ting

Supervisor: Dr. Reynold C.K. Cheng

**April 15, 2018**

# *Table of Contents*

# 1 Abstract

The graduate program admission process for HKUCS requires a great deal of effort and time for the admission officers every year. However, no solid records of admission strategies applied in previous years are found to serve as a reference readily available. As a result, the decision makers must correlate the information of candidates and results manually, which records the information inefficiently. The project aims at enhancing the existing analyzing system with the parameterized multimedia data of the candidates. The project introduces several techniques to convert the multimedia data into parameter-like data, and integrates existing data with the parameterized multimedia data. The insight of the whole set of candidates' data is expected to present in the final report of the project. Users could select factors on the web-based presentation tool to understand the influence on admission result of each factor. Further studies and implementation could be applied to other admission data of other departments and schools, or on company hiring process.

# 2 List of Figures

# 3 List of Tables

# 4 Abbreviations

| Abbreviated Phrases | Original Phrases |
|---|---|
| HKU | The University of Hong Kong |

| | |
|---|---|
| HKUCS | Department of Computer Science, Faculty of Engineering,<br>The University of Hong Kong |
| The Department | Department of Computer Science, Faculty of Engineering,<br>The University of Hong Kong |
| MPhil | Master of Philosophy |
| Ph. D. | Doctor of Philosophy |
| CGPA | Cumulative Grade Point Average |

Table 1: Abbreviation

# 5 Introduction

## *Background*

Plenty of applications for graduate programs, such as MPhil and Ph. D., are received by the Computer Science Department of the University of Hong Kong (HKUCS) every year. Because of the limited spaces for candidates, manually selecting the suitable candidates takes much effort and time. Apart from that, the Department has also to take the probability that the selected candidates may reject the offer into account. Therefore, providing insightful advice for the selection of the candidates and finding out the key attributes of the admission patterns provide the decision makers, admission officers and professors, an effective way to admit the most suitable candidates.

In the last year, an analysis system, which allows users to interact with the result pattern on a web-based interface, has been constructed. The users could choose the combinations of the factors of the candidates' application data, which are mainly Boolean and parameter data, such as CGPA, English test results, and their research interests. In addition, the project is divided in three parts. First, the system extracts the information from the database where all the admission data is stored. Next, it analyzes and aggregates the data with data-mining algorithms, and obtain the insights of selecting successful candidates. Finally, the system visualizes the result with the factors chosen in diagrams.

## *Objectives*

To gain a deeper understanding of the admission criteria, apart from the result of the parameter data analysis, the current project aims at analyzing multimedia formats of data, such as interview videos, teacher's comments and resume contents. The multimedia data has been collected for years and it is available for analysis. By using data mining tools and multimedia analyzing techniques, the application can provide a more thorough analysis of the admission patterns. With the help of parameterized multi-media, the project has a better suggestion of the appropriate students to the decision-makers according to the students' research interests as well as the prediction of the result of prospective students. In short, the goals are listed below:

- Multimedia data parameterizing transformation
- Analysis of parameterized multimedia data with the different models
- Enhancement of web-based interface showing data presentation and classification result

## *Related Studies*

Educational data mining can take advantage of the learning system to identify certain patterns among the collected parameter and multimedia data. For instance, some studies used models to carry out data analysis task, such as classification, regression, density estimation correlation mining and sequential pattern mining. [1] Also, descriptive and predictive approach techniques are improved to better adapted to the focused data format. Cluster analysis groups data based on similarities, logistic regression and decision tree enhance the accuracy of the predictive approach. [2]

Scarano et al. developed speech recognition methods for audio data analysis and data mining. Spoken words can be searched as an organized data format. [3] Additionally, the application can organize the data with an SQL-like interface for processing, searching and combining with other traditional data formats. With statistical and clustering processes, recent studies have developed a realistic pattern of educational data mining approaches. [4]

Thomas M. developed techniques to present the multimedia data as informative factors [5] and Ramous J has vectorize the importance of words in documents with statistical methods [6]. The methods are very helpful to extract useful information from the multimedia format data.

# 6 Methodology

## *Tools for Data Transformation and Analyzing*

### *1. Transformation tools for the multi-media data*

- Google Speech Recognition API [7]: an online speech-to-text tool provided by Google, but with 15 seconds limitation for free services
- Pocket Sphinx [8]: an offline Python package allows user to train the speech recognition module with customized input

- Python speech features library [9]: for extracting lexical, acoustic, fluctuation and other main acoustic features from audio files

### 2. *Text-to-Vector transformation tools*

- Bag of words: A method use hashing for mapping all the words in a text file into a hash table and record the appearance time of the words. The "words" are divided and extracted from the strings in the document with a space for English. It disregards the orderings of the words. Examples

  [Document 1]: I major in software engineering, and I have some publications in software engineering.

  [Document 2]: Peter has done some projects in software engineering.

| | I | Major | In | Software | Engineering | And | Have | Some | Publications | Peter | has | done | Some | Projects |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| D2 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

- Bigram Bags of words: This is the extension of bag of words. It concatenates the next words and records the concatenation's appearance time. For example, the two documents become:

  [Document 1]: (I, major). (major, in), (in, software), (software, engineering), (and I), (I, have), (have some), (some publications), (publications in), (in software), (software engineering).

[Document 2]: (Peter has), (has done), (done some), (some projects), (projects in), (in software), (software engineering)

- TFIDF [6]: term frequency-inverse, it is a method to represent the importance of the words in a document or a corpus. If a word is repeatedly appearing in a document, the value indicating the importance will increase proportionally. Example:

  [Document 1]: Mary join one project.

  [Document 2]: Mary join another program.

  Tf("Mary", Document 1) = 1 / 5 = 0.2

  Tf("Mary", Document 2) = 1 / 7 = 0.14

  Idf("Mary", Document 1 & Document 2) = log (2/2) = 0

  Tfidf("Mary", Document 1) = 0.2 * 0 = 0

  Tfidf("Mary", Document 2) = 0.14 * 0 = 0

  While "Mary" in two documents are having two importance 0, it indicates "Mary" does not have much importance in the both documents.

  Tf("program", Document 1) = 0 / 5 = 0

  Tf("program", Document 2) = 3 / 7 = 0.429

  Idf("program", Document 1 & Document 2) = log(2/1) = 0.301

  Tfidf("program", Document 1) = 0

  Tfidf("program", Document 2 ) = 0.429 * 0.301

- Word2Vec: [5]

  Word2Vec is a training model with two layers of neural network with a large pre-trained corpus, which can identify the word as word vectors of position proximity in a large dimensionality. The numeric presentation can project the relativity of the words. If pairs of words have similar relativity of meaning, the difference of the words in each pair are similar.

### 3. *Data analyzing tools*

- MySQL connector: Python library to retrieve the tables and data stored in MySQL database management system. Users modify the MySQL database with Python codes through MySQL connector.
- Scikitlearn [10]: machine learning library for the Python programming language
  Classifiers under Scikitlearn: BernoulliNB classifier, RandomForestClassifier, Regressor, LogisticRegression, MLPClassifier
- NumPy [11]: a Python library with a large collection of mathematical functions aiming at processing large, multi-dimensional arrays and matrices, which is used to aggregate the parameterized multimedia data
- Pandas [12]: a Python package with expressions of data structure allows users to work with relational and labeled data, used to sort and categorized parameterized multimedia data
- Tensorflow [13]: an open source library that offers data mining models to build and train with labeled and arbitrary data
- ROC curve [14]: A fundamental tool for evaluating test result. The true positive rate is plotted with the relative proportional false positive rate.

### 4. *Database system access and storage form*

- MySQL & MySQL workbench: application data of graduate program admission is stored in MySQL database management system.

- JSON & CSV file: two kinds of semi-formatted file, usually for data storage. The previous parameter data is stored both in JSON and CSV file, because they are both compatible to import to and export from the MySQL database system. The two kinds of files are supported by the database management system, easy to be restored or backed up when migrating the system.

### 5. *Web interface construction*

- D3.js [15]: JavaScript library for producing dynamic, interactive data visualizations on web interface using SVG, HTML5, and CSS standards
- HTML/CSS: HTML defines the structures of webpages, and CSS represents the layouts of webpages

### 6. *Video Parameterization tools*

- Affectiva [16]: API supports several features of the face. It recognizes emotional states in faces which is consists of each emotion’s portion, such as sadness, neutral, disgust, anger, surprise, fear, contempt, and happy. Also, the emotions and the appearance can be evaluated with the designed time frame.

The figures provided below are the web interface produced by previous students. Multimedia factors are added into the choices, and the patterns are integrated with their analysis result.

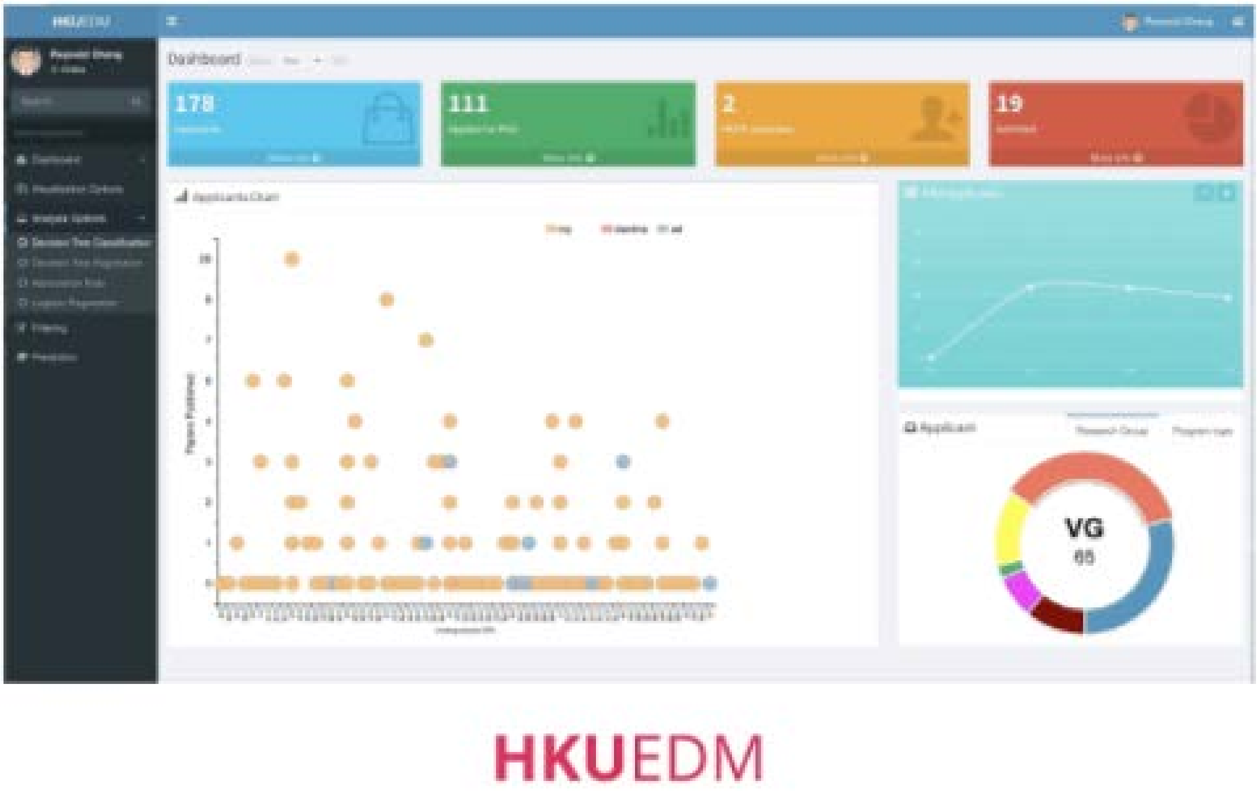
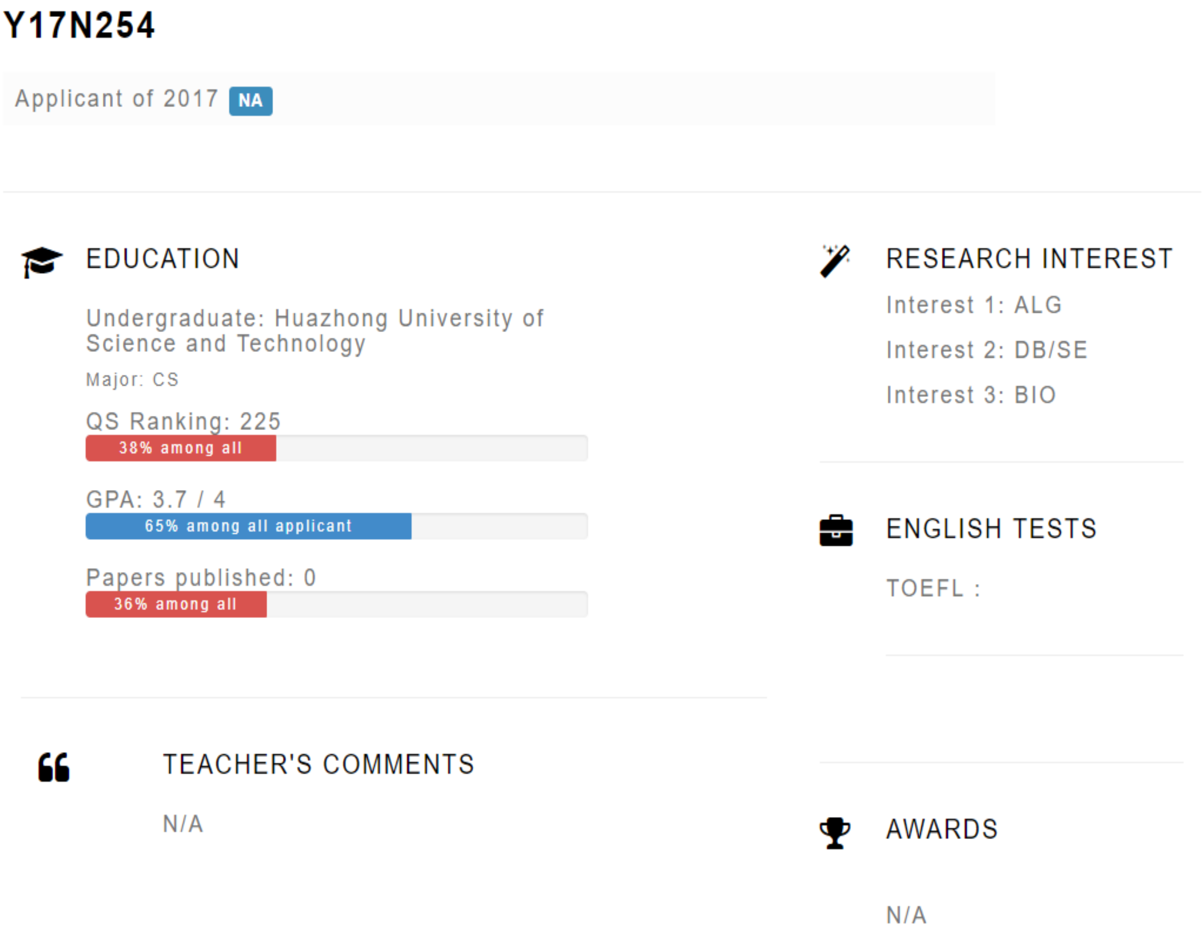


*Figure 1:Dashboard with Different Data Visualization Widgets by Previous Students* [17]

Figure 1 illustrates the dashboard on the web-based interface. It shows the correlation of selected factors on diagrams. Users get deepened understanding by reading the demographic statistics of the selected factors.



Y17N254

Applicant of 2017 NA

EDUCATION

Undergraduate: Huazhong University of Science and Technology

Major: CS

QS Ranking: 225

38% among all

GPA: 3.7 / 4

65% among all applicant

Papers published: 0

36% among all

RESEARCH INTEREST

Interest 1: ALG

Interest 2: DB/SE

Interest 3: BIO

ENGLISH TESTS

TOEFL :

TEACHER'S COMMENTS

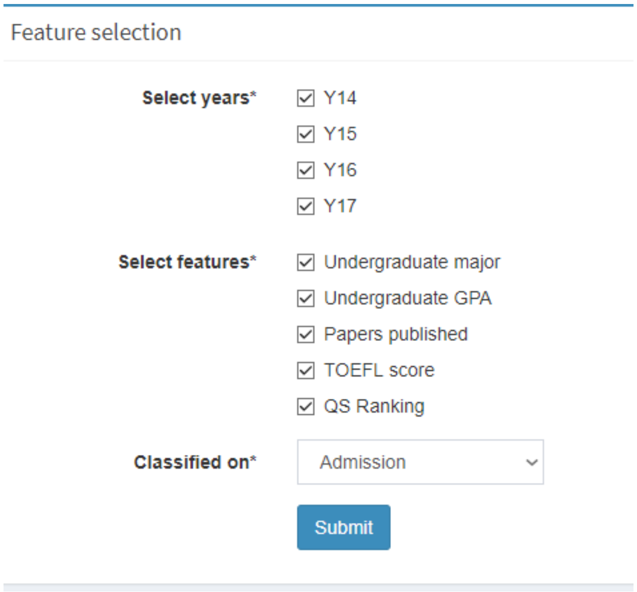
N/A

AWARDS

N/A

*Figure 2: Website Showing the Application Detail by Previous Students* [13]

Figure 2 provides the application details of one candidate. On the information page, users get knowledge about the overview of student's academic information.



Feature selection

Select years*
☑ Y14
☑ Y15
☑ Y16
☑ Y17

Select features*
☑ Undergraduate major
☑ Undergraduate GPA
☑ Papers published
☑ TOEFL score
☑ QS Ranking

Classified on* Admission

*Figure 3: Feature Selection*

Figure 3 shows the selection feature of factors. The web-based interface provides available factors to be selected, and users indicate the combination of factors such as admission year, undergraduate major and GPA, and classification method. Multimedia factors are added on the list of features to be selected with smart filtering algorithms applied. Tentative matching data and suggestion are appeared after the selection.

Also, the multimedia part of the analysis is integrated into the web application. If the candidate has a skype interview or a face-to-face interview, users can view the related files and analysis result with the different function on the web application.

# *System Design*

## *7. Database Management System and Storage*

The admission process of HKUCS includes academic information submission and interview session with admission officers. The numerical and Boolean types of academic information have been collected and stored in a JSON and CSV data formats, which allows the users to reconstruct the database if needed. The current database system is in MySQL, and the schema of the database are available on MySQL Workbench.

MySQL database management system ensures the data security and provides the scalability of the data. The type of the data is predefined when the tables are created in the database. Also, SQL has a clear standard of the data format to be stored for each entry, which safeguards the consistency of the data inside the tables. Consequently, there is no invalid or unrelated data in each predefined column because the data is categorized when it is inserted in the table.

The grant of data integrity helps the aggregation done in the analysis step, which requires consistent and sorted. In addition, MySQL database management system restricts the access of databases, so any unauthorized user has no permission to modify the schema of the tables inside. In addition, SQL quires retrieve many records from a database efficiently on the volume of data stored in the project.

On the other hand, the analyzed multimedia data is stored in JSON file, and it can choose to analyzed with the numerical database.

## *8. Acquisition of Multimedia data*

During the interview sessions, all the questions to the interviewees and the response of the interviewees are recorded and stored. From the interview video, the facial expressions and the audio information are extracted and processed. The performance of every candidate, based on the facial expressions and the audio information, in their interview, are examined and analyzed for making implications of the admission strategies of HKUCS.

The main data sources, the videos recorded of the interviewees are a reliable reference of the candidates since the videos contain the real-time interactions between the candidates and the admission officers. The videos and the audios have not been modified or compressed. Therefore, the reactions and responses in the videos are authentic and are ideal for feature extractions.

## *9. Aspects Chosen on Analysis*

Previously, the analysis model of Boolean type and parameter data is constructed. The first step of the admission pattern of the applications is based only on the model of Boolean type and parameter data. On the other hand, the multi-media data has not yet been processed and analyzed. The audio files and video files are stored in the formats when the videos are recorded during the interviews and on the online application system.

The video and audio files contain real-time interactions between the interviewers and applicants, which are lack and inconceivable in the model built in the previous project. Moreover, a variety of reactions and expressions of the applicants can be found in the multimedia files, such as facial expressions, speech tones and emotions. Therefore, in order to obtain a complete picture of the underlying admission patterns, it is necessary to examine multimedia data.

The information could be retrieved from video data can be categorized into three types: [18]

(i) Low-level feature information: color, texture, and shape

(ii) Syntactic information: noticeable objects, objects' temporal position, and spatial relations between objects

(iii) Semantic information: descriptive events happening perceivable, and interactions between actors. For example, spatial aspect of the characters and the movements of tracking objects

In the video files recorded, the camera focused on candidates' faces and has never moved. Therefore, the information contained in the videos is the voice of the interviewers and candidates, candidates' postures, movements and other features on the faces. As the three categories defined, the low-level feature information to be extracted are the color, texture, and shape of the environment of the interview site. The syntactic information are objects appeared in the video, such as candidates face and chairs. In the projects, semantic information is mainly focused, which refers to the events happening between the candidates and the interviewers.

The interview videos are the main data source for this project. The video can be divided into two parts, video files and audio files. In the video files, since the faces of the candidates are the main actors, the three levels of information extracted in order are color, texture and shape of the faces (skin tones, features of facial organs), movements and spatial relationship between the objects on their faces (face recognition), and movements and interactions on the faces (facial expressions). Assumed skin tones, features of facial organs are unrelated to the admissions, i.e. interviewers do not judge on appearance and are not racists, and facial recognition is irrelevant since the name of the video is marked with candidates' names and only one face is appeared in each video, facial expressions are the focus of the information to be extracted from the video files.

The information extracted from audio files are categorized into two kinds of aspects [19]: Text-based indexing and phoneme-based indexing. Text-based indexing converts speech to text and identifies words with entries, such as speech recognition. Phoneme-based indexing converts speech to phonetic text and only works with sounds, such as tones, intonations, and fluctuations. Emotions in conversations are extracted in the audio files. With the linguistic features, accents and pronunciations of the talker are available. In short, the content of conversations between interviewers and candidates are extracted with speech recognition tool and phonetic texts are extracted with emotions and linguistic recognition tools.

In conclusion, the available and useful data from the video and audio files are the facial expressions on the candidates' faces, speech tones, and phonetic scripts.

### *10. Multi-media Data Transformation*

The project focuses on two types of information: facial expression and speech analysis. Facial expressions are extracted and analyzed by a facial expression recognition tool, Affectiva API and Node.js library. Speech analysis are processed with tools on speech recognition, speech tone extraction, and linguistic features, with Tensorflow-speech-recognition, Conceptor, and Accent-classifier.

## Visual Data Processing

To extract the facial expression, the program utilizes Affectiva API analyzes the content in the extracted facial expression. The web application is implemented with Javascript along with Node.js and Ajax, in which the videos are cut into a series of photos and send to the API. With the large set of the photos produced by the application, each of photo is scanned to obtain features such as portions of every kind of facial expressions, such as happiness, sadness, anger, disgust, fear, contempt, surprise, and neutral. The percentage of each facial expression in each photo is summed up and the arithmetic average portions represent the portions of emotions in the video. Then, the percentages of videos are calculated for further analysis.

Figure 4 illustrates the series of photos cut from the video. Each photo represents different snapshot under a timeframe. The combination of different emotions is produced by each facial recognition result of photo.

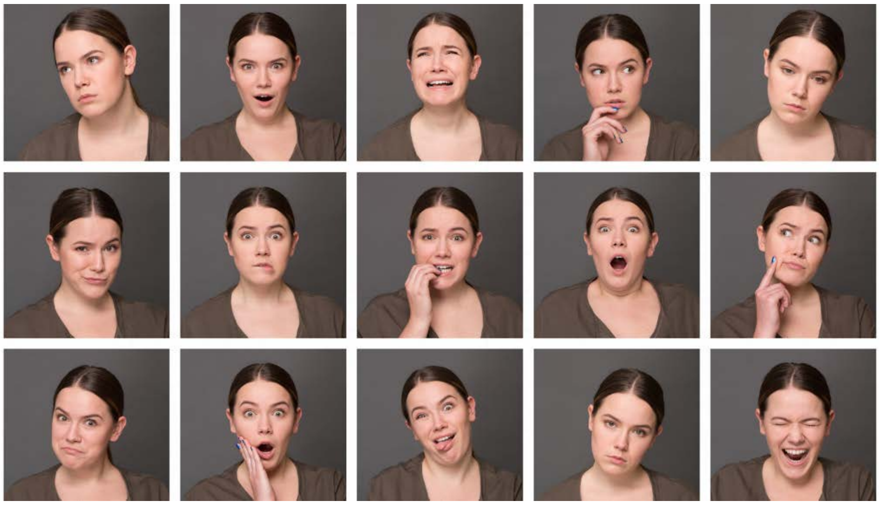
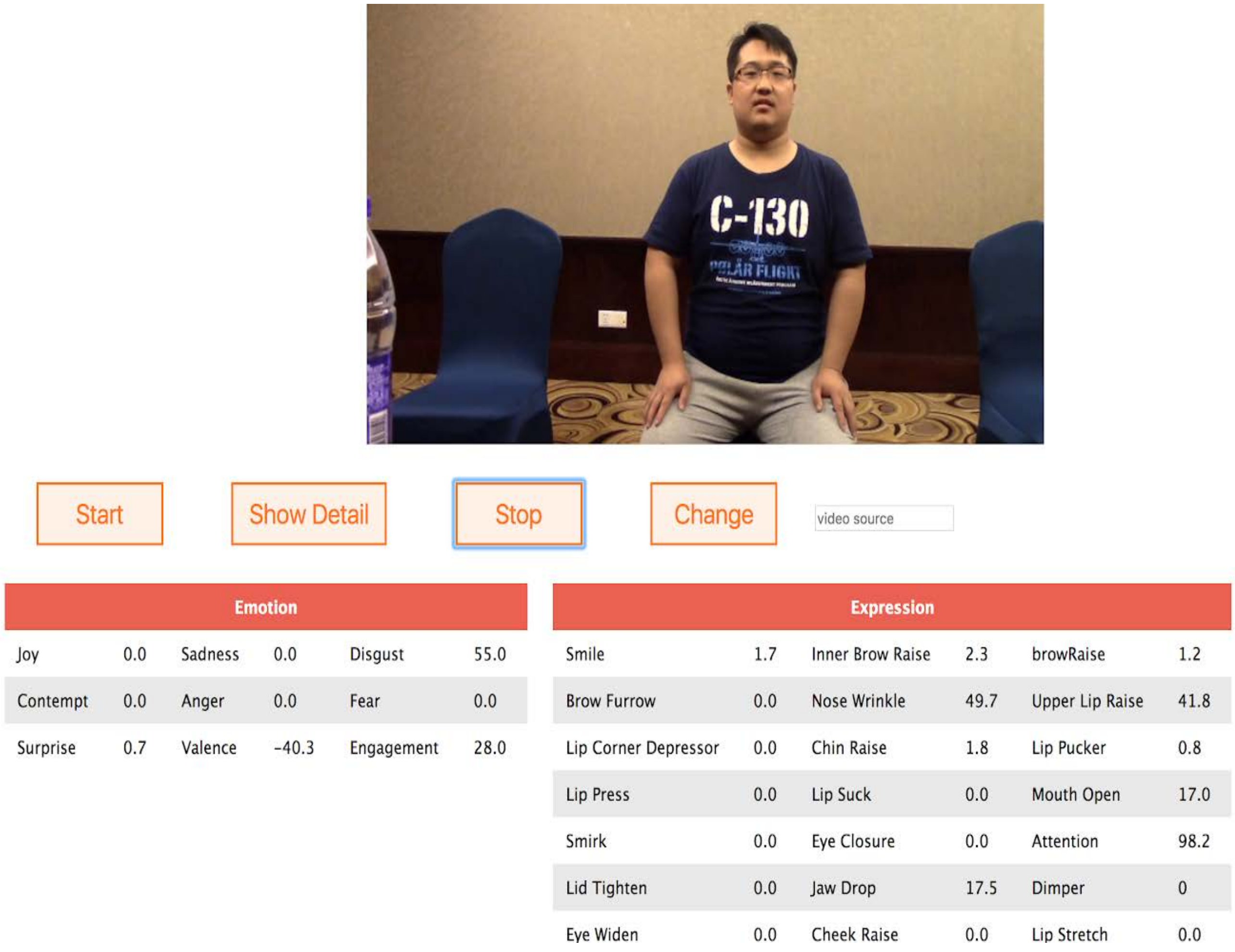


*Figure 4: Recognizing emotional states in faces*

The web application is served as an interface for user to input the file and get the numerical result with support for a variety of operating systems, such as Windows, Linux, Mac, iOS, and Android. Since the photos are cut in a constant time interval in the video, the photos account for same importance weight of a video. Hence, the arithmetic average of each emotion's percentage represents the emotion composed of a video.



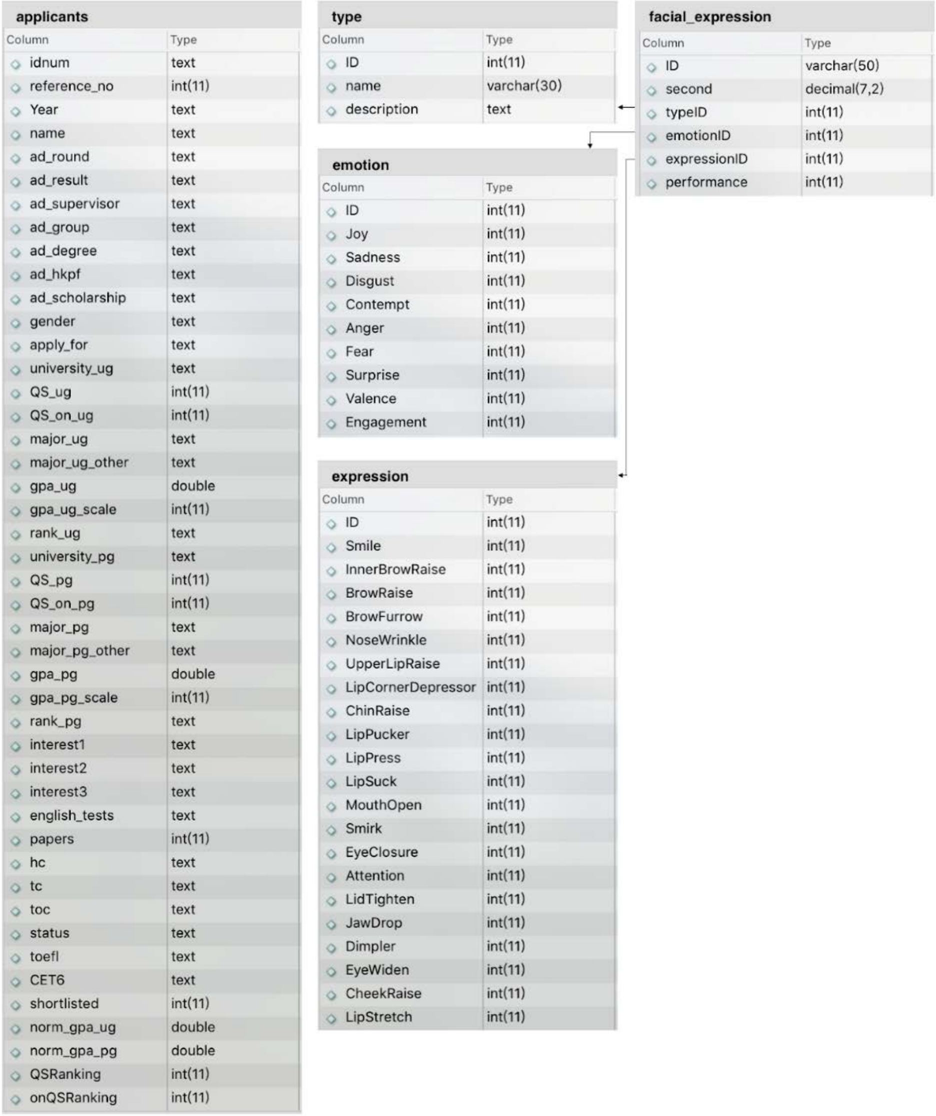
| Emotion | | | | | |
|---|---|---|---|---|---|
| Joy | 0.0 | Sadness | 0.0 | Disgust | 55.0 |
| Contempt | 0.0 | Anger | 0.0 | Fear | 0.0 |
| Surprise | 0.7 | Valence | -40.3 | Engagement | 28.0 |

| Expression | | | | | |
|---|---|---|---|---|---|
| Smile | 1.7 | Inner Brow Raise | 2.3 | browRaise | 1.2 |
| Brow Furrow | 0.0 | Nose Wrinkle | 49.7 | Upper Lip Raise | 41.8 |
| Lip Corner Depressor | 0.0 | Chin Raise | 1.8 | Lip Pucker | 0.8 |
| Lip Press | 0.0 | Lip Suck | 0.0 | Mouth Open | 17.0 |
| Smirk | 0.0 | Eye Closure | 0.0 | Attention | 98.2 |
| Lid Tighten | 0.0 | Jaw Drop | 17.5 | Dimper | 0 |
| Eye Widen | 0.0 | Cheek Raise | 0.0 | Lip Stretch | 0.0 |

*Figure 5: Affectiva API with frontend presentation*

Users can use web applications to play the selected video and obtain the corresponding parameterization of the analyzing feature straightforwardly. Also, the time frame of slicing video is able to be defined by users.

The variables and extracted categorizes are plotted as the following figure.



**applicants**

| Column | Type |
|---|---|
| idnum | text |
| reference_no | int(11) |
| Year | text |
| name | text |
| ad_round | text |
| ad_result | text |
| ad_supervisor | text |
| ad_group | text |
| ad_degree | text |
| ad_hkpf | text |
| ad_scholarship | text |
| gender | text |
| apply_for | text |
| university_ug | text |
| QS_ug | int(11) |
| QS_on_ug | int(11) |
| major_ug | text |
| major_ug_other | text |
| gpa_ug | double |
| gpa_ug_scale | int(11) |
| rank_ug | text |
| university_pg | text |
| QS_pg | int(11) |
| QS_on_pg | int(11) |
| major_pg | text |
| major_pg_other | text |
| gpa_pg | double |
| gpa_pg_scale | int(11) |
| rank_pg | text |
| interest1 | text |
| interest2 | text |
| interest3 | text |
| english_tests | text |
| papers | int(11) |
| hc | text |
| tc | text |
| toc | text |
| status | text |
| toefl | text |
| CET6 | text |
| shortlisted | int(11) |
| norm_gpa_ug | double |
| norm_gpa_pg | double |
| QSRanking | int(11) |
| onQSRanking | int(11) |

**type**

| Column | Type |
|---|---|
| ID | int(11) |
| name | varchar(30) |
| description | text |

**emotion**

| Column | Type |
|---|---|
| ID | int(11) |
| Joy | int(11) |
| Sadness | int(11) |
| Disgust | int(11) |
| Contempt | int(11) |
| Anger | int(11) |
| Fear | int(11) |
| Surprise | int(11) |
| Valence | int(11) |
| Engagement | int(11) |

**expression**

| Column | Type |
|---|---|
| ID | int(11) |
| Smile | int(11) |
| InnerBrowRaise | int(11) |
| BrowRaise | int(11) |
| BrowFurrow | int(11) |
| NoseWrinkle | int(11) |
| UpperLipRaise | int(11) |
| LipCornerDepressor | int(11) |
| ChinRaise | int(11) |
| LipPucker | int(11) |
| LipPress | int(11) |
| LipSuck | int(11) |
| MouthOpen | int(11) |
| Smirk | int(11) |
| EyeClosure | int(11) |
| Attention | int(11) |
| LidTighten | int(11) |
| JawDrop | int(11) |
| Dimpler | int(11) |
| EyeWiden | int(11) |
| CheekRaise | int(11) |
| LipStretch | int(11) |

**facial_expression**

| Column | Type |
|---|---|
| ID | varchar(50) |
| second | decimal(7,2) |
| typeID | int(11) |
| emotionID | int(11) |
| expressionID | int(11) |
| performance | int(11) |

*Figure 6: Features extracted with Affectiva API*

### Audio Data Processing

Concerning speech recognition, the projection extracts mainly focus on the content of speech. After the Sphinx training for the neural network or connecting with the Google Speech Recognition API, the content of the interview are obtained. Next, if the candidates have multiple interviews, the contents of multiple files are combined into one, representing all the speech in the interviews.

After getting the speech content in the interview, the words in the content file are evaluated and transformed into numerical formats. Bag of words, bigram bag of words, tfidf, and word2vec represent the words or groups of words with numbers to indicate the importance or the words or group of words. Therefore, the documents of the interview content are transformed into weights indicating the importance of each word compared to all the files of interview content.

Google Speech Recognition API has a relatively higher accuracy of the transcribed text compare to PocketSpinx. With the vectorizing methods of bags of words, bigram bag of words, tfidf, and word2vec, the content of speech in the interview sessions are successfully represented as a numerical format comparing to the documents.

Then, the relative admission result of the candidate corresponding the document is retrieved. We split the sample for training and sample for testing with a 3:1 proportion. Finally, we choose the classifier to identify the admission result of the testing sample with the prediction method gained from the training session.

On the other hand, during the training session, the variables of the information of the candidates can be freely chosen. Other numerical information, such as length of the speech and interview, can also be added to the training.

After the processing of the data transformation, the parameterized data is stored in the current database as parameter data, which is compatible with the current storage on the database management system.

A large set of databases aiming at data analysis are available in Python, such as NumPy, Scikilearn, Pandas. Moreover, analyzing and extracting with Python packages are easily available to install. Additionally, Python has packages to connect to MySQL database management system, MySQL connector, and supports many functions to create a web service to enable other people to retrieve and interact with analysis result.

### *11. Data Analysis*

Before the classification the parameterized data, the tools must be able to retrieve the data from the database management system. First, the MySQL connector must be implemented. With the data collected and stored in the database management system, MySQL, the analyzing tools retrieve the parameterized data.

Next, the application will adopt the appropriate model and framework to store the data imported into the application. The model is then trained with part of the imported labeled data and construct a classification for the data chosen. After the training of the model, the rest of the data is used to examine the rate of the correctness of the model.

Figure 7 illustrates the decision trees the Python data analysis libraries have found. It classifies the candidates into eight categories, using four levels of classifications. It divides the candidates with GPA factor twice, and different kinds of categorizations follow. Users can now choose the added feature of multimedia for analysis, if the candidates have the interview session during the application process.

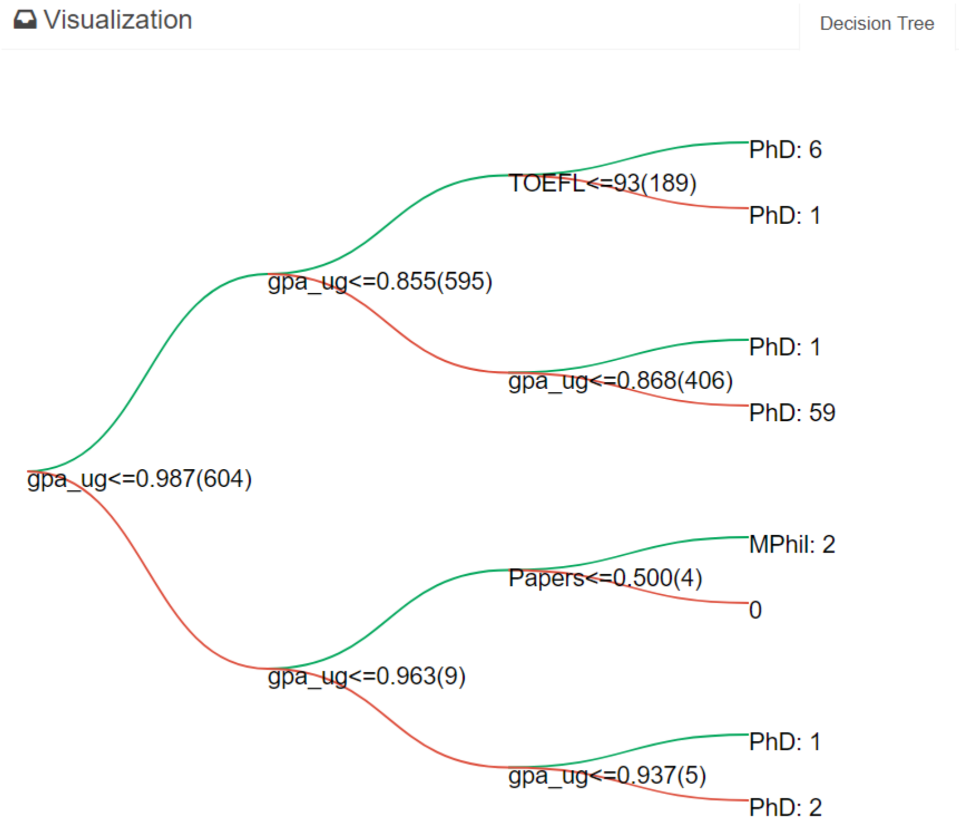


*Figure 7: Four Levels of Candidate Categorization [13]*

## *12. Web-based Interface*

The web tool developed by the previous students [16], Wu You and Xu Fang Yuan, presents the result of the analysis of the updated database, with the Boolean type and parameter type data and the parameterized video and audio data. Apart from the current factor selection tools, the project increases the options for the parameterized multi-media formats. Users are free to choose the combination of the Boolean and parameter data and the multi-media formats. As a result, the web-based interface facilitates more the data presentation and the classification result with the help of enhanced web tool.

The web-based interface allows the observers to access and interact with the database system, and it is a user-friendly environment that provides clear choices of the factors. The implementation with the analysis result is also easily available with the use of D3.js application.

Figure 8 gives an example of correlation graph of selected features. Users get knowledge of factor correlation with the labels provided.

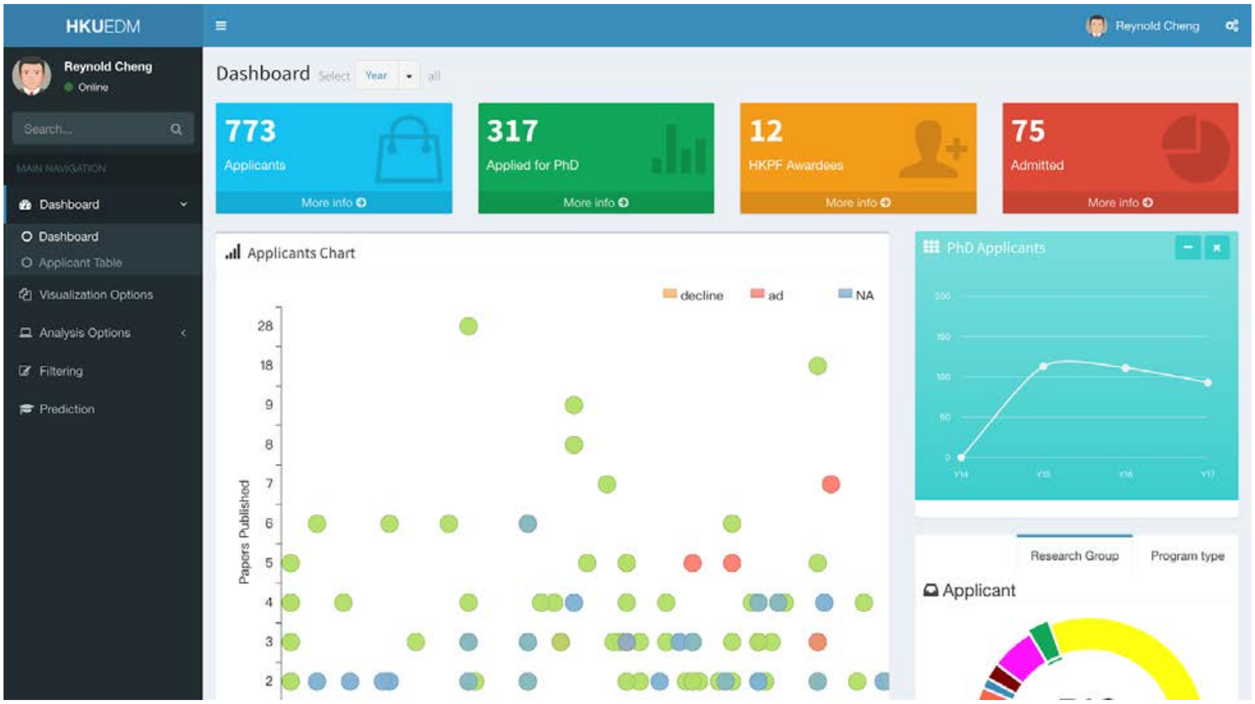


*Figure 8: Feature Selection*

# 7 Result and Evaluation

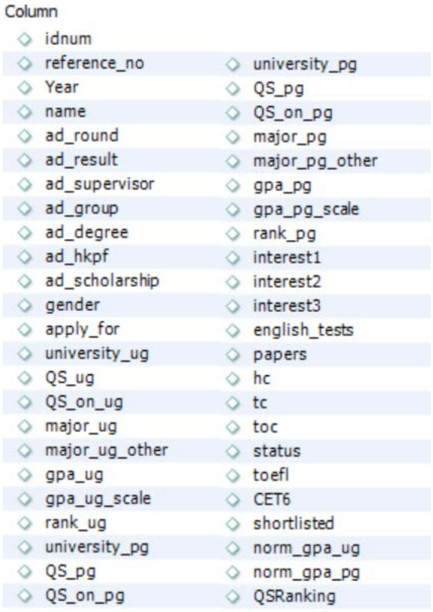
## 1. *Package Installation and System Setup*

The essential Python packages, such as Pandas, Tensorflow, NumPy, Scikitlearn have been installed on both Windows and Ubuntu/Linux environments. MySQL, and MySQL Workbench, which is applied as database management system, have been set up on the environments. Node.js and the browsers have been installed on the environments. The Django framework web-based tool has been re-installed and working on the environments.

## 2. *Database Construction and Schema*

On MySQL Workbench, table visualizations are provided, and users can see the relevant datasets as a result of queries. Figure 8 gives a glimpse of candidates' information stored in the database, containing gender, program applied, undergraduate major, and English test result. Apart from the Boolean and parameter data produced by the previous project, additional columns dedicated to parameterized multimedia data formats are added. For example, each category of facial expressions, each indicator of accent group, texts of interview audio file content, and emotion percentages contained in video and audio files.

Since additional columns are added to each candidate, the schema of tables is modified. Tables for each perspective of parameterized multimedia is constructed, and they are linked to the table of candidates' basic information, for the system to track each candidate's information.



Column

- idnum
- reference_no
- Year
- name
- ad_round
- ad_result
- ad_supervisor
- ad_group
- ad_degree
- ad_hkpf
- ad_scholarship
- gender
- apply_for
- university_ug
- QS_ug
- QS_on_ug
- major_ug
- major_ug_other
- gpa_ug
- gpa_ug_scale
- rank_ug
- university_pg
- QS_pg
- QS_on_pg
- university_pg
- QS_pg
- QS_on_pg
- major_pg
- major_pg_other
- gpa_pg
- gpa_pg_scale
- rank_pg
- interest1
- interest2
- interest3
- english_tests
- papers
- hc
- tc
- toc
- status
- toefl
- CET6
- shortlisted
- norm_gpa_ug
- norm_gpa_pg
- QSRanking

*Figure 9: Original Scheme of Query Result on MySQL Workbench*

### 3. *Speech Extraction Tools*

Feature extractions on speech consist of speech recognition, speech accent classification, speech tone extraction, and speech to text techniques. For each kind of extraction, specific applications have been chosen to complete the tasks as follows:

| Technique | Decision | Description |
|---|---|---|
| Speech recognition | Speech Recognition application with Python library – Pocket Sphinx | Provides a library for performing speech recognition, supporting several engines and API.<br>It classifies the speaker of the speech with libraries provided. Also, users can freely choose and train the model with customized data, and it is efficient to be implemented on Python application. |
| Speech to text technique | Tensorflow-Speech-Recognition | Tensorflow-Speech-Recognition tool is a Python speech recognition tool on deep learning framework, Tensorflow.<br>It is trained with a large data set including synthetic Text to Speech snippets, Movies with transcripts, Gutenberg, YouTube with captions, with some extensions available. |

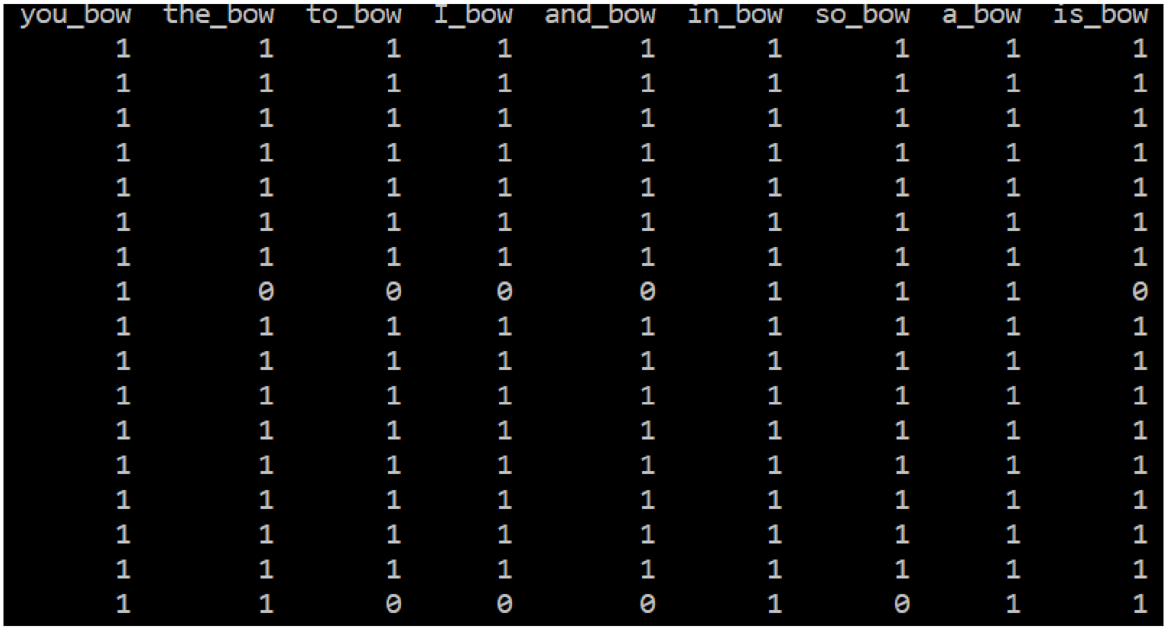
*Table 2: Choices of Speech Feature Extraction*

### 4. *Speech Content Parameterization*

After obtaining the speech content of the interviews, the program needs to evaluate the speech content, with the vectorization of the speech content. There are several methods to convert the words in the documents into evaluable numerical presentations. Then, the parameterization model is combined with the corresponding admission result, which is used to train for the prediction result.

Apart from the accuracy rate directly obtained compared to the true admission result, the program also utilizes the ROC curve coverage area to visualize the true positive rate to the false positive rate.

*Bag of words*

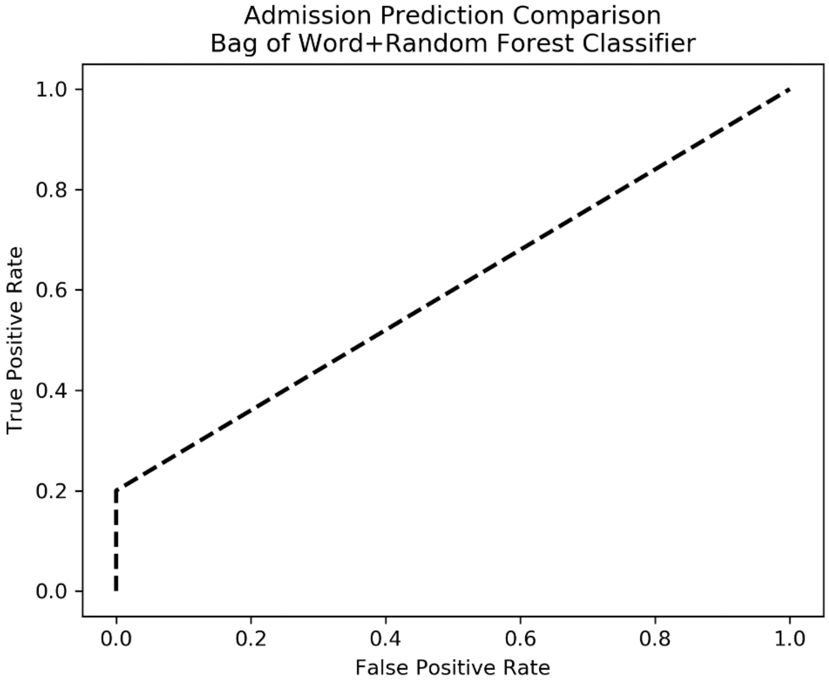


```
you_bow  the_bow  to_bow  I_bow  and_bow  in_bow  so_bow  a_bow  is_bow
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        0       0      0        0       1       1      1       0
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       1      1        1       1       1      1       1
      1        1       0      0        0       1       0      1       1
```

*Figure 10: Bag of word models for training and testing*

First, the program scans all the words containing in all the documents and store the appearing words in a list. Second, each word in a document are scanned with hashing function, map to the table if the sentence contains the words. Then, the table are added with corresponding information, such as document length and video file length, and the table is merged with admission result to form the data model to train and test. The train-to-test ratio is 3:1.

The true positive rate to false positive rate are plotted as below. The performance is skewed to the true positive rate, which means it performs better with the positive prediction. The accuracy rate with random forest classifier is 0.85 with ROC area covered 0.85, and the random forest regressor holds an accuracy rate of 0.88 with ROC area covered 0.6.



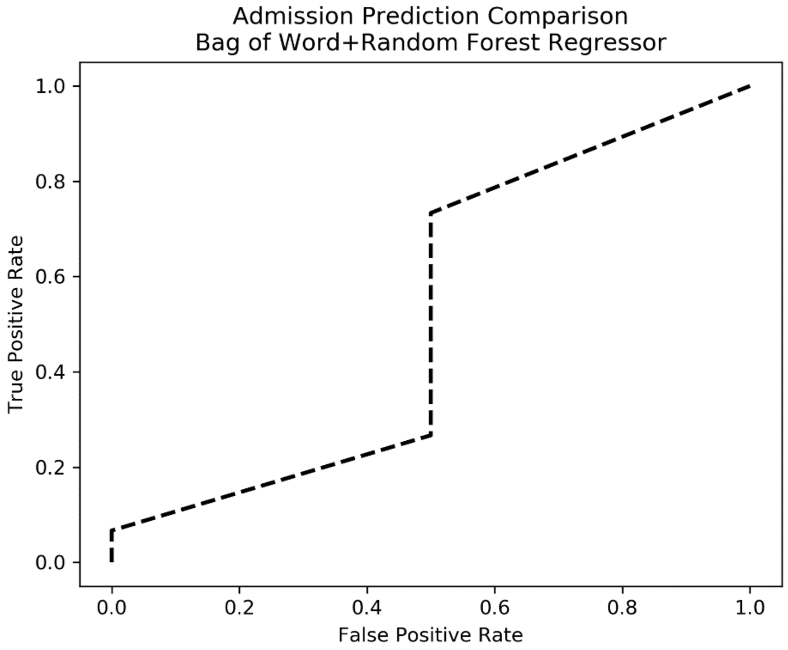


*Figure 11: Bag of word models accuracy of prediction result*

*Bigram bag of words:*

When the program scans through all the document and record the words, it groups the next word of words into a bag, treating it as the word in bag of words. Then, it adds the previous processed bag of words into the variables, leading the words appearing in two or more documents have more appearance records of all the documents. However, due to the limited documents obtained from the interviews, the bigram bags of words have a poor performance since few documents hold the same bigram bags in common.

The following are predicted with random forest classifier and random forest regressor. More than 90 percent of documents have the same prediction result. Therefore, the resulting diagrams appear diagonal and both accuracy rates are 0.85.

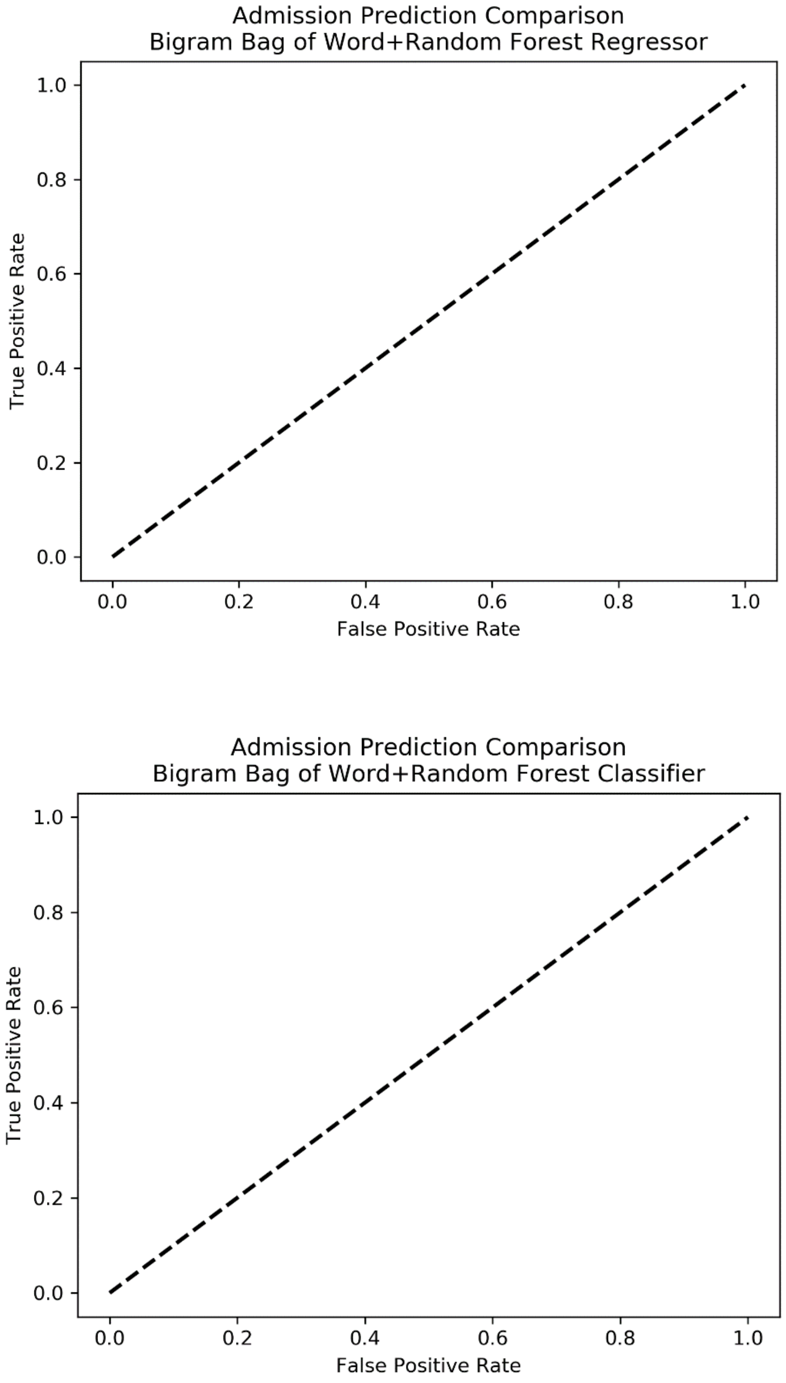
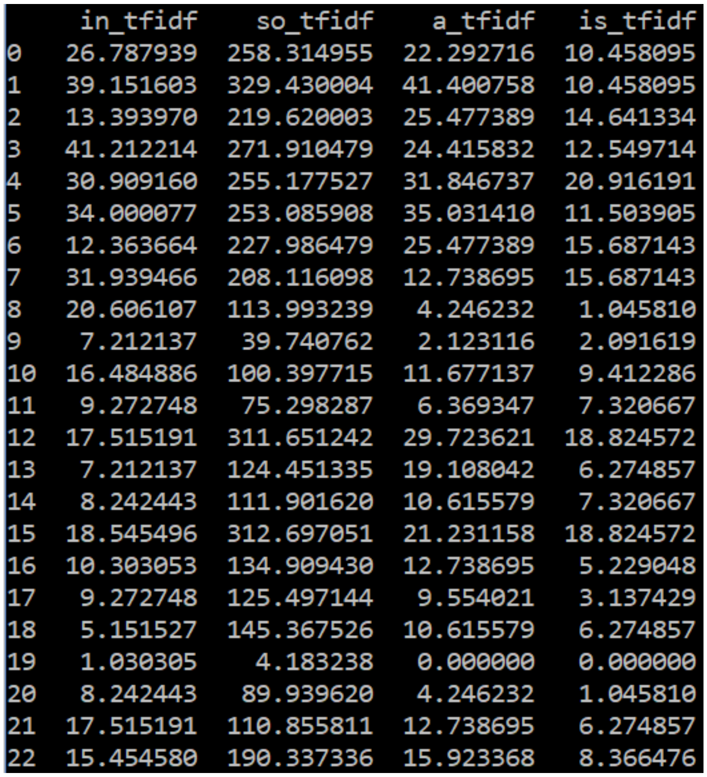


*Figure 12: Bigram bag of word model’s accuracy of prediction result*

*TFIDF (Term frequency – inverse document frequency):*

After the program scans through all the documents and record the appearance of the words in all the documents. Next the weighting of the words is assigned with the term frequency times the inverse document frequency. If word A appears in every document, the weightings of word A in each document are zero, because the inverse document frequency is all zero. On the other hand, if a word B only appears in half of the documents, it has a higher weighting for the documents where word B appears, and still maintain 0 for the documents where word B disappear.



```
     in_tfidf    so_tfidf    a_tfidf   is_tfidf
0   26.787939  258.314955  22.292716  10.458095
1   39.151603  329.430004  41.400758  10.458095
2   13.393970  219.620003  25.477389  14.641334
3   41.212214  271.910479  24.415832  12.549714
4   30.909160  255.177527  31.846737  20.916191
5   34.000077  253.085908  35.031410  11.503905
6   12.363664  227.986479  25.477389  15.687143
7   31.939466  208.116098  12.738695  15.687143
8   20.606107  113.993239   4.246232   1.045810
9    7.212137   39.740762   2.123116   2.091619
10  16.484886  100.397715  11.677137   9.412286
11   9.272748   75.298287   6.369347   7.320667
12  17.515191  311.651242  29.723621  18.824572
13   7.212137  124.451335  19.108042   6.274857
14   8.242443  111.901620  10.615579   7.320667
15  18.545496  312.697051  21.231158  18.824572
16  10.303053  134.909430  12.738695   5.229048
17   9.272748  125.497144   9.554021   3.137429
18   5.151527  145.367526  10.615579   6.274857
19   1.030305    4.183238   0.000000   0.000000
20   8.242443   89.939620   4.246232   1.045810
21  17.515191  110.855811  12.738695   6.274857
22  15.454580  190.337336  15.923368   8.366476
```

*Figure 13: TFIDF for training and testing*

The TFIDF methods is very useful for filtering out the common words which appear in every document, such as “I”, “you”, “and”, “or”, “not”. It does not require manually filtering the common words after the training, and it miss no words appearing in all the documents. However, the program can perform the pre-training common word filtering, which enhances the content of the documents with abandoning the meaningless common words.

With the effective filtering, the random forest classifier has given out a good prediction rate with 0.82 accuracy and a ROC area of 0.8.

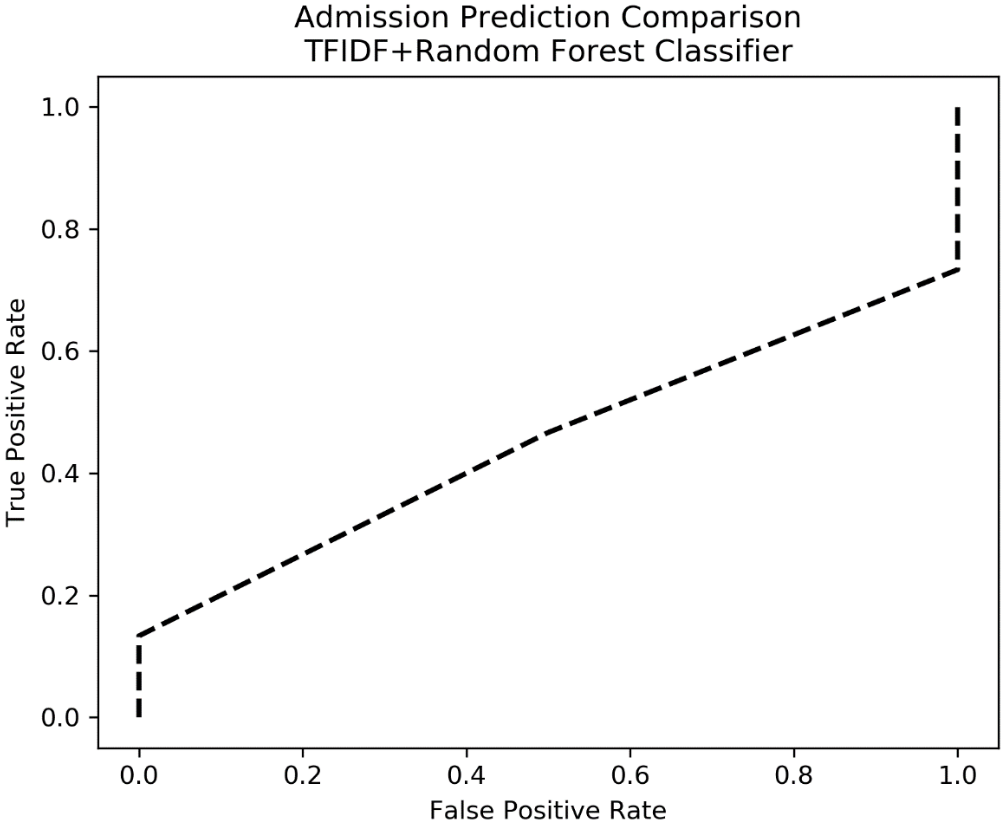


*Figure 14: TFIDF model’s accuracy of prediction result*

However, with random forest regressor, the ROC area is lower with the plot skewed to the false positive rate, which means the random forest regressor gives out a higher probability of false positive prediction. It only covers ROC area of 0.55, with an accuracy rate of 0.88.

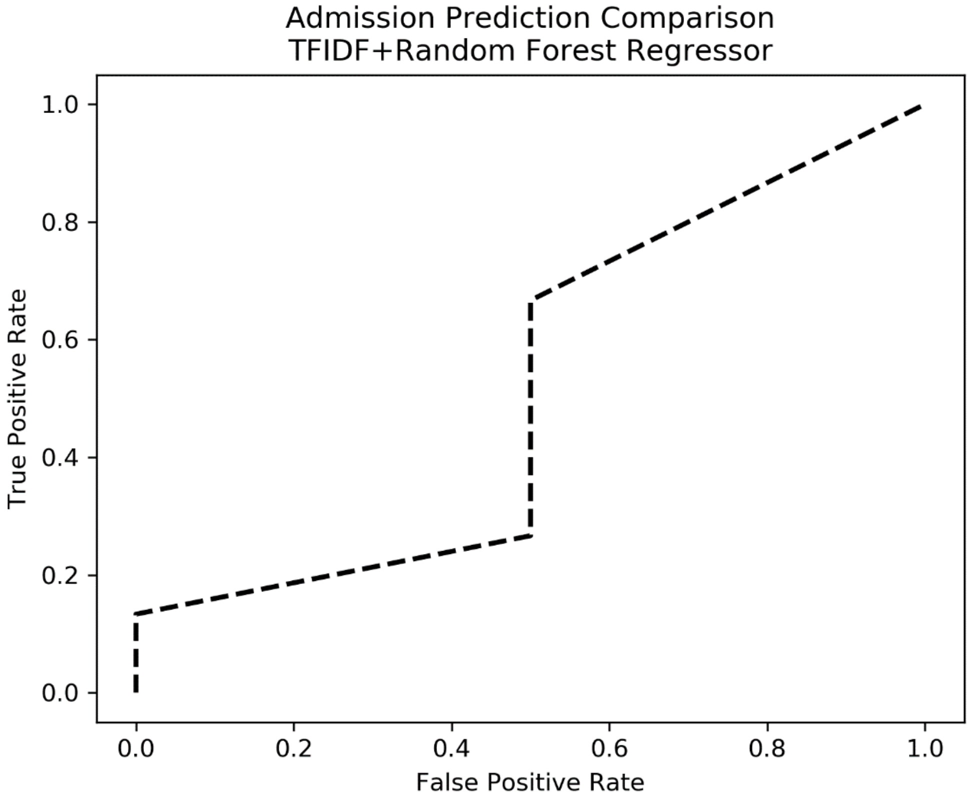
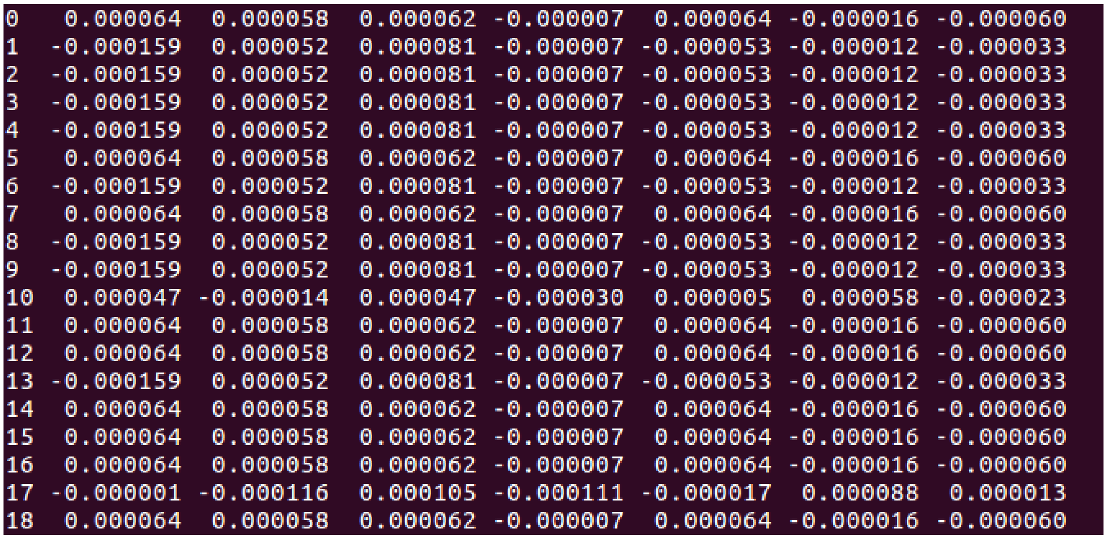


*Figure 15: TFIDF model's accuracy of prediction result*

*Word2Vec*

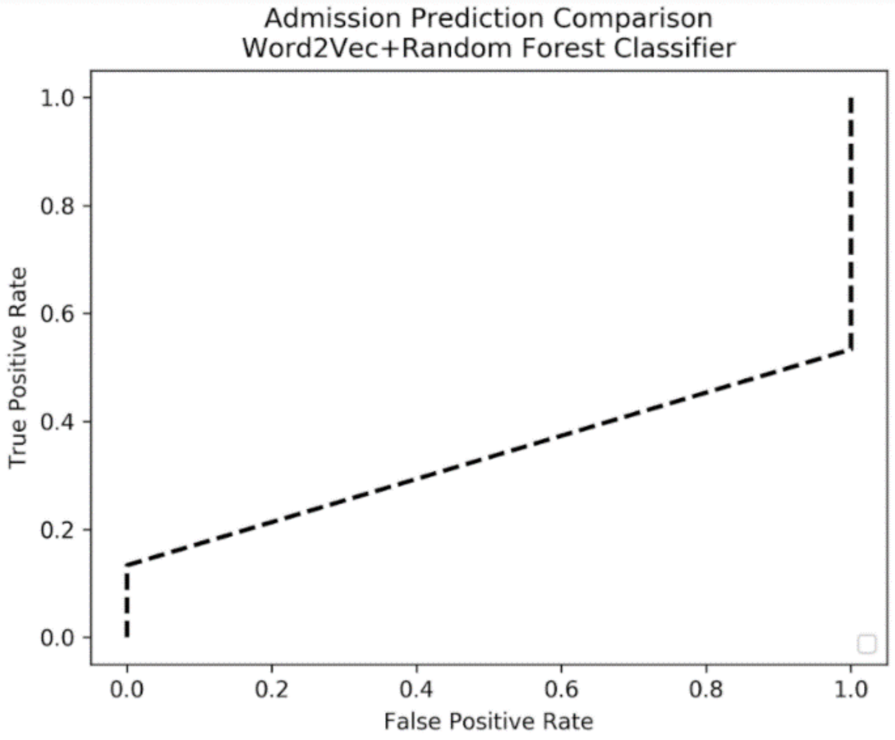
Word2Vec takes a large input of words corpus and produce relativities of the words with a large dimensionality. Here, the program takes all the documents as the large set of words are a corpus and obtain the relativities of the words in these documents. The program first retrieves the projected relativity of the word and combined according to sentences. After 4 iterations, the combining positions of the words in documents are obtained with values:



```
0    0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
1   -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
2   -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
3   -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
4   -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
5    0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
6   -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
7    0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
8   -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
9   -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
10   0.000047 -0.000014  0.000047 -0.000030  0.000005  0.000058 -0.000023
11   0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
12   0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
13  -0.000159  0.000052  0.000081 -0.000007 -0.000053 -0.000012 -0.000033
14   0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
15   0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
16   0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
17  -0.000001 -0.000116  0.000105 -0.000111 -0.000017  0.000088  0.000013
18   0.000064  0.000058  0.000062 -0.000007  0.000064 -0.000016 -0.000060
```

*Figure 16: Word2Vec for training and testing*

After getting the model, the program split the sample and the test portions for prediction. With random forest classifier and random forest regressor, the accuracy rates of the prediction are both 0.82 with ROC curve coverage area of 0.73 and 0.66. The accuracy is not surprisingly high, and the ROC curve coverage area is lower than TFIDF with random forest classifier. However, the Word2Vec parameterization method yields a strong difference to the binary classification. When Bag of words or TFIDF methods give a low variation of the predictions, Word2Vec has the highest maximum difference of the predictions.



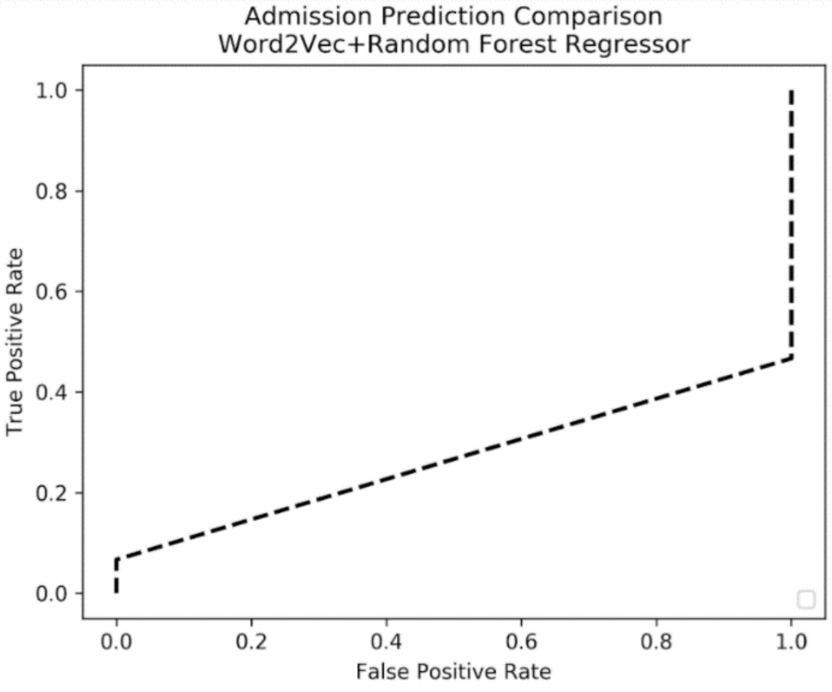


*Figure 17: Word2Vec model's accuracy of prediction result*

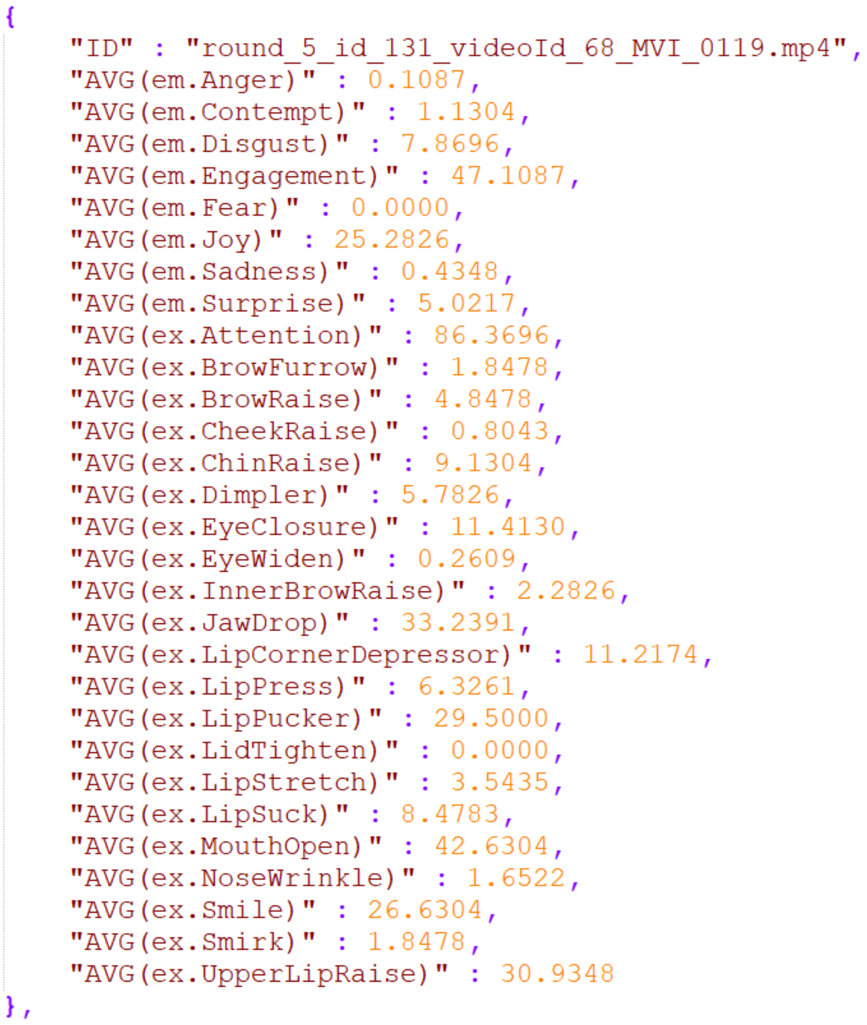
### 5. *Facial Expression Tools*

| Decision | Description |
|---|---|
| Microsoft Azure API | Microsoft offers an API for users to detect emotions expressions and cognitive services online. However, it requires upload of the candidates' file, causing concerns of privacy issues. |
| Affectiva | Also, it is compatible among the platforms. The application is setup as a web application with JavaScript SDK with HTML, CSS and JavaScript. The frame of web application is presented with HTML with styles formatted with the corresponding CSS file, and functionalities with the cross-platform JavaScript library. Also, the program utilizes jQuery and Ajax to facilitate the communication with the JavaScript library.<br>Plus, the back-end server for data-processing tool are constructed with Node.js and Express. Node.js supports the functionality with run-time programming functions, and Express serves as server-side scripting system. They provide a great compatibility which also allows users to perform task on different platforms. |

*Table 3: Facial Expression Tools*

The collected videos are uploaded to the web application based on Affectiva API and Nodejs package. The web application extracts the features as informative factors while the videos are cut into small chunks of video according to the time frame.

The web application uses Node.js and Express as backend server and the Affectiva JavaScript SDK as frontend server. The frontend server displays the current information factors with the most updated result computed with the backend server. For each second, the screenshot of the video is processed as an independent image sent to the Affectiva API, which later returns the values of the image feature it is responsible to detect.



```
{
    "ID" : "round_5_id_131_videoId_68_MVI_0119.mp4",
    "AVG(em.Anger)" : 0.1087,
    "AVG(em.Contempt)" : 1.1304,
    "AVG(em.Disgust)" : 7.8696,
    "AVG(em.Engagement)" : 47.1087,
    "AVG(em.Fear)" : 0.0000,
    "AVG(em.Joy)" : 25.2826,
    "AVG(em.Sadness)" : 0.4348,
    "AVG(em.Surprise)" : 5.0217,
    "AVG(ex.Attention)" : 86.3696,
    "AVG(ex.BrowFurrow)" : 1.8478,
    "AVG(ex.BrowRaise)" : 4.8478,
    "AVG(ex.CheekRaise)" : 0.8043,
    "AVG(ex.ChinRaise)" : 9.1304,
    "AVG(ex.Dimpler)" : 5.7826,
    "AVG(ex.EyeClosure)" : 11.4130,
    "AVG(ex.EyeWiden)" : 0.2609,
    "AVG(ex.InnerBrowRaise)" : 2.2826,
    "AVG(ex.JawDrop)" : 33.2391,
    "AVG(ex.LipCornerDepressor)" : 11.2174,
    "AVG(ex.LipPress)" : 6.3261,
    "AVG(ex.LipPucker)" : 29.5000,
    "AVG(ex.LidTighten)" : 0.0000,
    "AVG(ex.LipStretch)" : 3.5435,
    "AVG(ex.LipSuck)" : 8.4783,
    "AVG(ex.MouthOpen)" : 42.6304,
    "AVG(ex.NoseWrinkle)" : 1.6522,
    "AVG(ex.Smile)" : 26.6304,
    "AVG(ex.Smirk)" : 1.8478,
    "AVG(ex.UpperLipRaise)" : 30.9348
},
```

*Figure 18: Facial Features extracted by Affectiva API*

The most important character in the image is the face continuously detected in the images. The Affectiva API accumulate the value of the face features. When the data is accumulated more than the threshold set by the users, the jQuery API is triggered with an Ajax call to send back the feature factor result. Next, Express at the backend server is triggered to route the feature information interacting with different RESTful requests. When the backend system obtains the result, it re-route the result object with the file storing routers to the MySQL database, where the feature information is inserted with the MySQL queries. Last, the front-end displaying interface is updated with the newly obtained result.

## *Limitation and Mitigation*

The conversion and extraction of the multimedia data is a challenge for the project. Multimedia data, such as video and audio data, contain a great amount of information. It is crucial to choose the most relevant data among the whole set of candidates' data to improve the admission strategies. Also, integrating the newly built analyzing tool to the existing analyzing tool is hardly feasible, while multimedia data requires vectorization before being analyzed.

### *Limited Result Format*

Furthermore, the transformed data by the tools provided limits the result format. Because the existing database only contains the Boolean and parameter data, the transformation of the multimedia data also has to be parameterized, in order to be a part of the candidates' columnized data. In addition, the data analysis model for the current system only supports the parameterized data. Therefore, the limitation of the data parameterizing models is that the tools should be able to produce the suitable parameterized data for the database and data analysis use.

For example, the facial recognition tool, Affectiva gives out the numerical percentage of emotions in each picture frame. Tensorflow speech recognition gives out the transformation in text format. To find the boundaries of classifications, data analyzing tools must be adaptable to the numerical percentage results and text format. Bag of words, bigram bag of words, TFIDF, and Word2Vec give out the numerical weight or position of each word in documents. Without the labelling of each word in the dictionary, it is impractical to retrieve the tones and implications by the words. However, such mapping and labelling requires much work; thus it is always proprietary. Thus, the result format of data transformations is limited to the built and available tool, and the classifying tools chosen must be able to efficiently categorize the vectorization of the multimedia data.

*Financial Support*

On the other hand, the financial support is limited for this project. Expensive commercial tools provided by corporates are not affordable to be used. As a result, only open source packages and tools are available. Affectiva, Tensorflow, Python Speech Recognition tool are all open source projects and their codes are readily available. Also, the available online APIs charge with fees when the uploaded audio file is too long. Therefore, to cope with the problem, the program slice the audio file into small chunks, and combine the result to obtain the whole transcription of the audio file. This is not efficient and effective, while the speech recognition mainly depends on the continuity of linguistic sounds

*Low Accuracy of Data Transformation*

During the interviews, the candidates are usually in a tight mood and do not answer the question fluently as prepared. On the other hand, the interviewers are very confident and pronounce the prepared questions loudly and clearly. Therefore, the accuracy of the result of speech recognition has a strong difference between candidates and the interviewers. The speech recognition tools usually obtain as a clear and comprehensible sentence for interviewers, whereas the content of candidates' speech and answers are blurred with wrong words.

On the other hand, the model trained depends fully on customized data. English speech recognition data are usually trained with British or American accents. However, in our speech files, the candidates are mostly Chinese, which has difference pronunciation and accents

*Analysis Models Chosen*

To analyze the whole set of the data with the multimedia data converted, the program chooses the binary classification and group the testing data into [admitted, not admitted]. The classification tool can independently evaluate the multimedia data successfully. On the other hand, in the previous work, a model is chosen for the Boolean and parameter values. Because the classification and regression tools are not compatible with the previous built system, the program has to produce a result of multimedia first and integrate with the existing candidates' information

*Privacy issues regarding online analyzing tools*

Apart from the limitation mentioned above, since the data source is from HKUCS, it could neither be published or be uploaded to any cloud service. The data should be kept confidential. Therefore, some cloud service is only for reference. The safe way to parameterize the multimedia data is to construct offline tools, such as train the Sphinx tool with customized data. Nevertheless, compared too the existing online tool, the offline self-made tools often hold a slower speed and poorer accuracy rate

*Inconsistent information format*

During the different years, the schemes of applications changes. The features and information obtained from candidates are different from year to year. Therefore, the analysis and format of different year has different results.

# 8 Conclusion and Future Planning

## *Conclusion*

The main implementation of the project is the development of a toolbox extracting useful information from the multimedia. The program has extract and parameterize the aspects extracted from multimedia data. In contrast, the extractions and parameterization are fairly defined by the tools chosen. Accordingly, the project assists admission officers to adopt better admission analysis strategy to the future admission decision. The department members and other admission decision-makers can have a deeper understanding of many aspects of the candidates with the well-constructed data mining methods. Also, the prospect students can follow the predictions by the deliverables constructed and make the best application choice. We hope the developed application could be adopted for candidate selection from other departments at HKU, or be an applicable strategy for company hiring process.

## *Future Planning*

After the data extractions from multimedia data format and transformation, in the future, the following works can be done:

- Since the data analysis tool and data mining algorithm is developed, it can be utilized for text-based data, audio data and video data.

- The projects can be applied for other application process, different department, or the recruitment process of commercial hire.

# 9 References

[1] Romero C., & Ventura S. (2010) Educational Data Mining: A Review of the State of the Art. IEEE Trans. Syst., Man, Cybern. C. 40: 601-618.

[2] Vijayakumar, V. & Nedunchezhian, R. (2012) A study on video data mining 1: 153. https://doi.org/10.1007/s13735-012-0016-2

[3] Scarano, R., & Mark, L. (2006). U.S. Patent No. US 7133828 B2. Washington, DC: U.S. Patent and Trademark Office.

[4] Alejandro P. (2014) Educational data mining: A survey and a data mining-based analysis of recent works. Journal of Educational Data Mining. Volume 41, Issue 4, Part 1

[5] Tomas M. (2013). Distributed Representations of Words and Phrases and their Compositionality, arXiv: 1310.4546

[6] Ramos J (2003) Using TF-IDF to Determine Word Relevance in Document Queries

[7] Google Cloud Speech API [Internet]. Google Cloud Speech API. [cited 2018 Jan 28]. Available from: https://cloud.google.com/speech/

[8] Pocket Sphinx [Internet]. cmusphinx/Pocketsphinx. [cited 2018 Jan 28]. Available from: https://github.com/cmusphinx/pocketsphinx

[9] Python speech features. (n.d.). Retrieved from https://github.com/jameslyons/python_speech_features

[10] Scikit-learn [Internet]. machine learning in Python 0.18 documentation. . [cited 2018 Jan 28]. Available from: http://scikit-learn.org/stable/index.html.

[11] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011),

[12] Wes McKinney. Pandas: Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

[13] Martín Abadi (n.d.) TensorFlow: Large-scale machine learning on heterogeneous systems, (2015) Software available from tensorflow.org.

[14] Andrew P. (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms

[15] D3.js - Data-Driven Documents [Internet]. D3.js - Data-Driven Documents. Available from: https://d3js.org/.

[16] Danial M. (2013) Affectiv a-MIT Facial Expression Dataset (AM-FED): Naturalistic and Spontaneous Facial Expressions Collected "In-the-Wild"

[17] Wu, Y., & Xu, F. (2017, May). Mining HKUCS Graduate Student Data: Extraction, Analysis, and Prediction . Retrieved from http://www.cs.hku.hk/programme/projects/csfyp/csfyp.jsp

[18] Shirahama K, Ideno K, Uehara K (2005) Video data mining: mining semantic patterns with temporal constraints from movies. In: Proceeding of seventh IEEE symposium on multimedia, pp 598–604

[19] Surendra Shetty, K.K. Achary (2008) Audio Data Mining Using Multi-perceptron Artificial Neural Network