Chasing Your Dream School
Big Data Analytics Final Report

Xiuyue Wang (xw2454), Bing Han (bh2589), Chee Kit Tang (ct2819)
Biomedical Engineering, Electrical Engineering
Columbia University in the City of New York
xw2454@columbia.edu, bh2589@columbia.edu, ct2819@columbia.edu

Abstract

College is the most important phase of a life. Hence, choosing a right college is important. However, there is a lack of college recommendation system in the current market. To tackle this problem, we decided to build a web application that aims to recommend the school based on the prospective students' preference. The main concept of our application is to extract the relevant information from the big college data released by the U.S. Department of Education. In addition, our application will also predict the earning after graduation for the students to evaluate the value of pursuing a higher education. Through this application, we hope that the student will be able to find a suitable college and avoid the pitfall of suffering from college debts.

College; University; Education; Recommendation System; Big data; Earning Prediction

I. Introduction

In the past decade, there are a rising number of universities and many ranking system (Times Higher Education Ranking, QS World University Ranking and etc) aims to rank the colleges according to their professional standing. However, many existing ranking system choose to rank the school based on the number of research papers and citations[1] in addition to other factors. Hence, this irrelevant ranking system cannot be a useful metric for the students to make decisions.

Therefore, there is a need for a relevant college recommendation system for the prospective students as choosing a college can be a very difficult decision to make just by the student themselves. This is due to the fact that there are many factors to consider (such as SAT score, location and suitability) when choosing the right University.

Furthermore, there is a demand for a good recommendation system as it is important to choose a right college. According to the huffington post article[2], a college decision will not only affects the next four years of life, but it also influences the rest of your life.

In recent years, there are also many concerns about the values in going to the college due to rising tuition fee of the Universities. According to the CNN news article[3], Goldman Sachs mentioned that “Many students are better off not going to mediocre colleges -- ones that rank in the bottom 25% of all universities.They earn less, on average, than high school graduates.”

To prevent the students from wasting their time in education (when they earn lesser than high school graduates after their university graduation), our college recommendation system also aims to compute an earning prediction. This earning prediction is designed for the prospective students to consider whether is it worth to pursue a higher education.

II. Related Works

The current solution in recommending the colleges are based on the university ranking system as mentioned earlier. In addition to that, the prospective student usually consult their family, teachers and friends regarding their University application. However, personal opinions may be biased and restricted to a few choices. This is because there are up to thousands of colleges in the United States and it is impossible for a normal human being to provide an unbiased view from such a wide range of options.

In addition, there are also a few analysis studies done by individual colleges. However, their results may be skewed to convince the prospective students to join their school. It is difficult to judge the fairness of such analysis results.

III. System Overview

Our system design can be described as the following flowchart:

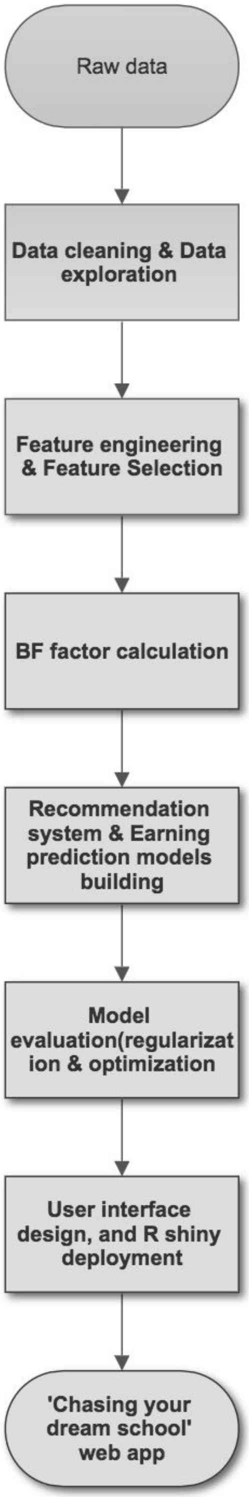

Raw data
Data cleaning & Data exploration
Feature engineering & Feature Selection
BF factor calculation
Recommendation system & Earning prediction models building
Model evaluation(regularizat ion & optimization
User interface design, and R shiny deployment
'Chasing your dream school' web app

Fig. 1. Flowchart of system design

Our raw dataset are downloaded from kaggle, it is the record of college scorecard ranges from 1996 to 2013. The size of our dataset is 1.3 GB. These data are provided through federal reporting from institutions, data on federal financial aid, and tax information. These data provides insights into the performance of schools that receive federal financial aid dollars, and the outcomes of the students of those schools.

After data cleaning and exploratory analysis, considering the inputs of our user interface, from hundreds of features in the dataset, we selected the following relevant parameters: Unit Id corresponding to each school, gender, ethnicity, household, household income, SAT score, locale and region. Furthermore, we also added tuition preference, SAT preference, locale preference, region preference. The preference is used to represent how important each feature is for our users. If the feature is really important in the evaluation criteria of our users, then we will increase the weight of this feature in the following calculation of Bayes factors.

Note that Bayes Factor (BF) is defined as ratio of the posterior-odds of the hypothesis. We will be using BF as the basis of our recommendation system and earning prediction method. More details will be introduced in the next section-Algorithm.

IV. Algorithm

Our whole system can be divided into two parts, the university recommendation part and the earning prediction part. The recommendation part is mainly based on applying the statistical analysis and distribution hypothesis on existing data. The earning prediction part is based on the Bayes Factor (BF) feature matrix and SVM classification model.

For the first part, the university recommendation, the main component is the bayes factor. The Bayes factor is the ratio of the posterior-odds of the hypothesis. For example, the probability after receiving the evidence, to the prior-odds of the hypothesis like probability before seeing the evidence. The evidence is the attribute defining a student cohort , like SAT score greater than 1400 , and the hypothesis is the attendance at the college.

For the computation for the feature corresponding to attribute (Attribute Y) for each college of interest (College X): we are using h denoting hypothesis, e denoting evidence, Cx denoting for college X and Ay denoting
for attribute y.

$$\begin{aligned} BF(h = C_X|e = A_Y) &= \frac{Odd(h = C_X|e = A_Y)}{Odd(h = C_X)} \\ &= \frac{P(e = A_Y)|h = C_X}{P(e = A_Y|h = NOT(C_X))} \\ &\approx \frac{P(e = A_Y)|h = C_X}{P(e = A_Y)}\} \end{aligned} \tag{1}$$

Fig 2. Bayes Factor Approximation

The last approximation portion is feasible because we have more than 1200 colleges in our working dataset, and hence, the student attending college is negligible.

Therefore, the probability of finding a student with Attribute Y amongst 1 students not at College X can be approximated as finding the same student amongst the entire student population (College X included).

Using this formula, we calculated the BF for ethnic, income, sat score, aid and etc. Some of the features is challenging. For example, for the SAT score, we need to get its approximate distribution, here we used the log normal distribution, based on the four quartile scores in the dataset.

Then we also added the BF indicator of location, including large city, small city etc,based on the site of the university. Another factor that we included is geographic preference, including Farwest, Mideast, Southwest and etc. The method used here is similar to the location.

For the recommendation system, we are using the aforementioned BF, based on the student profile, including student's sat score, family income and the preference for the university as well, and we calculated the final BF indicator for each college and get the highest N number of colleges as our recommendation.

The second part of this system is the earning prediction model. In this portion, we used the BF matrix as our feature matrix. Thereafter, we applied PCA to the BF features. Note that the original matrix included too many dimensions, for example, for SAT score feature, we had three columns with different range of sat score. Hence, there are correlations between different columns. In a result, we applied PCA to BF matrix and reduce dimension to be 6 with each column corresponding to an individual feature.

The next step of the earning prediction model is to split dataset to training set and test set. We decided the splitting ratio to be 0.9 and then we built the regression model on our training dataset.

As our data is non-linear, we decided to use SVM regression model, decision tree regression and random forest regression. According to each model, cross validation was applied to find the best hyperparameter.

V. Software Package Description

For our project, we utilize python for data preprocessing and exploration parts, also, R Studio and R shiny are utilized for the implementation of our algorithm and the design of the UI, respectively.

For python, packages ‘sqlite3’ and ‘pandas’ are employed, and for R Studio and R shiny, libraries ‘maggrittr’, ‘dplyr’, ‘ggplot2’, ‘e1071’, ‘leaflet’, ‘shiny’ are used.

‘sqlite3’

‘sqlite3’ is a C library that provides a lightweight disk-based database that doesn’t require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language.

Basically, by importing ‘sqlite3’, a database connection to the SQLite database can be build, and then we can download our data from the sqlite file, use SQL to manipulate the data and output the processed data into csv file for further use.

‘pandas’

‘pandas’ is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

For our project, we finished the data cleaning, feature engineering parts in python by using ‘pandas’.

‘magrittr’

For the implementation of our algorithm and our UI, we choose to use R and R shiny.

In our code, ‘magrittr’ is a package with two aims: to decrease development time and to improve readability and maintainability of code. In our code, magrittr provides a new “pipe”-like operator, %>%, with which we can pipe a value forward into an expression or function call.

‘dplyr’

‘dplyr’ is a package for data manipulation which provides some great, easy-to-use functions that are very handy when performing exploratory data analysis and manipulation.

‘ggplot’

ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts. It takes care of many of the fiddly details that make plotting a hassle as well

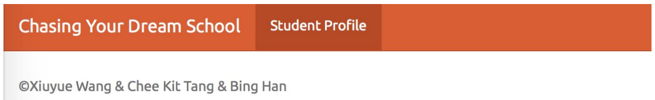
as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics.
In our project, 'ggplot' is basically used for plotting the outputs of recommendation results and earning prediction results.

'e1071'
This package provides functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier and etc.

'leaflet'
Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps.
In our project, 'leaflet' is basically used for plotting the outputs of map.

'shiny'
Shiny is an R package that makes it easy to build interactive web apps straight from R. Our User Interface is based on this package.

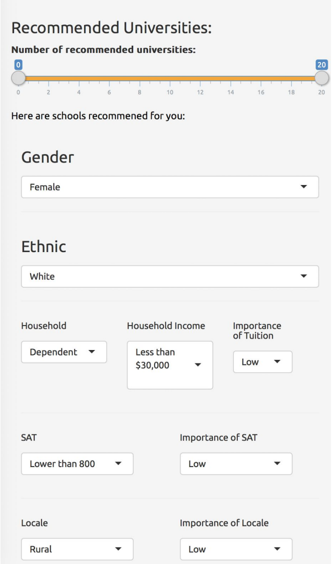
By using those packages mentioned above, we finally successfully designed a user-friendly and well-designed web app. Some screenshots are shown as below:


Chasing Your Dream School

Student Profile

©Xiuyue Wang & Chee Kit Tang & Bing Han

Fig. 3. Web app headline design


Recommended Universities:

Number of recommended universities: 0 – 20

Here are schools recommened for you:

Gender: Female

Ethnic: White

Household: Dependent

Household Income: Less than $30,000

Importance of Tuition: Low

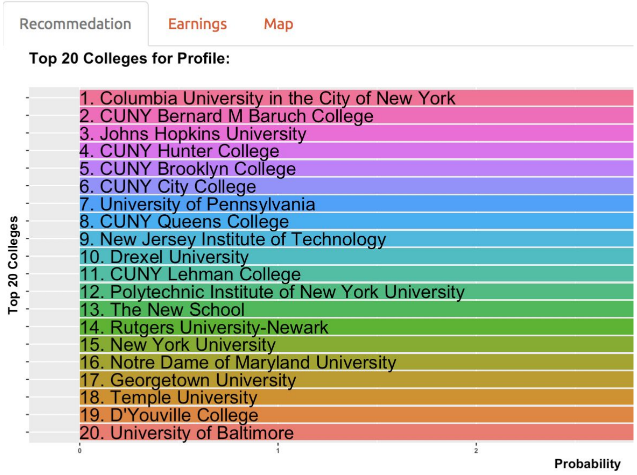
SAT: Lower than 800

Importance of SAT: Low

Locale: Rural

Importance of Locale: Low

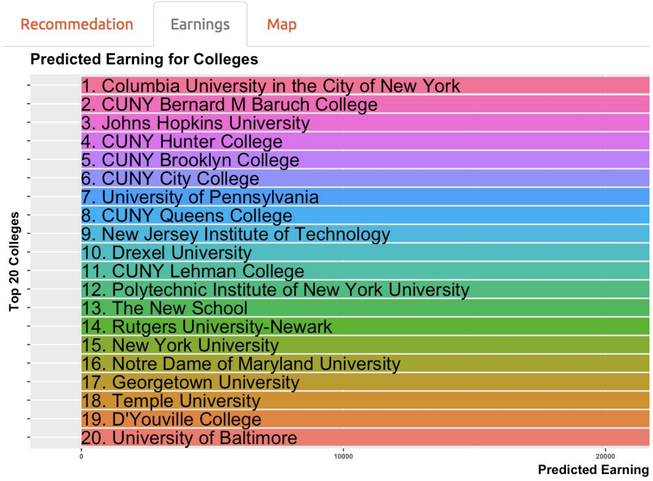
Fig. 4. Web app input panel



Top 20 Colleges for Profile:

1. Columbia University in the City of New York
2. CUNY Bernard M Baruch College
3. Johns Hopkins University
4. CUNY Hunter College
5. CUNY Brooklyn College
6. CUNY City College
7. University of Pennsylvania
8. CUNY Queens College
9. New Jersey Institute of Technology
10. Drexel University
11. CUNY Lehman College
12. Polytechnic Institute of New York University
13. The New School
14. Rutgers University-Newark
15. New York University
16. Notre Dame of Maryland University
17. Georgetown University
18. Temple University
19. D'Youville College
20. University of Baltimore

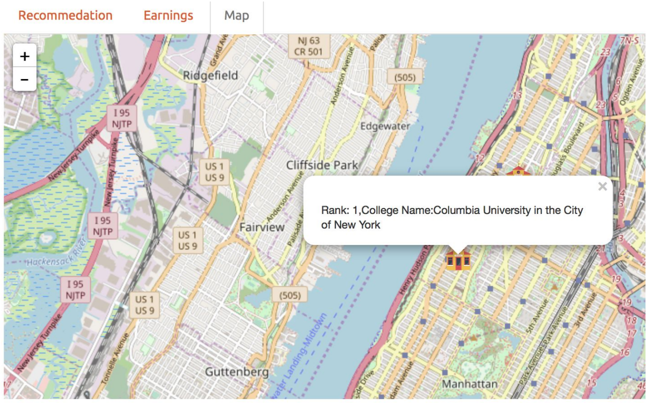
Fig. 5. Web app school recommendation results output



Predicted Earning for Colleges

1. Columbia University in the City of New York
2. CUNY Bernard M Baruch College
3. Johns Hopkins University
4. CUNY Hunter College
5. CUNY Brooklyn College
6. CUNY City College
7. University of Pennsylvania
8. CUNY Queens College
9. New Jersey Institute of Technology
10. Drexel University
11. CUNY Lehman College
12. Polytechnic Institute of New York University
13. The New School
14. Rutgers University-Newark
15. New York University
16. Notre Dame of Maryland University
17. Georgetown University
18. Temple University
19. D'Youville College
20. University of Baltimore

Fig. 6. Web app school earning predictions output



Rank: 1,College Name:Columbia University in the City of New York

Fig. 7. Web app locations of recommended schools output

VI. Experiment Results

Recommendation result

Based on the student profile and student's preference about the university, we calculated the BF indicator for each college and sorted them in descending order to get top n(n is the number of university user want to get) college. The results will display a bar graph as follow:

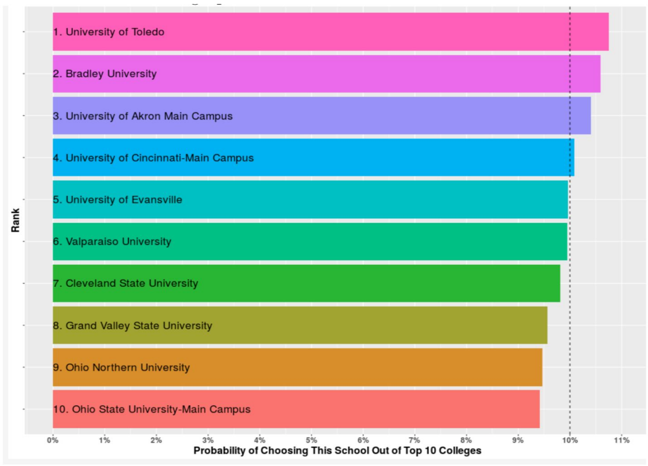
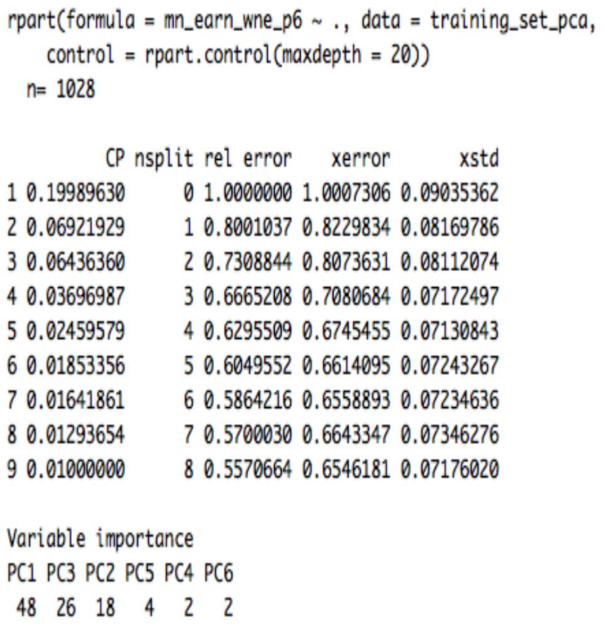


Fig. 8. Recommendation result

The evaluation part can be quite challenging for our system. This is because all of the features are based on University. We do not have any information from students such as rating or some implicit scores using in tradition recommendation model. Therefore, we decided to use another method to test our model. In this testing method, we randomly picked up 10 universities and found out 100 students who already were in this college and use their profile to test our model.

Earning prediction result

The prediction part are based on the BF matrix. In order to reduce the dimension and diminish the correlation between the features, we first applied PCA to BF matrix and reduced to six dimension. The result matrix is as follow:



```
rpart(formula = mn_earn_wne_p6 ~ ., data = training_set_pca,
    control = rpart.control(maxdepth = 20))
  n= 1028

          CP nsplit rel error    xerror       xstd
1 0.19989630      0 1.0000000 1.0007306 0.09035362
2 0.06921929      1 0.8001037 0.8229834 0.08169786
3 0.06436360      2 0.7308844 0.8073631 0.08112074
4 0.03696987      3 0.6665208 0.7080684 0.07172497
5 0.02459579      4 0.6295509 0.6745455 0.07130843
6 0.01853356      5 0.6049552 0.6614095 0.07243267
7 0.01641861      6 0.5864216 0.6558893 0.07234636
8 0.01293654      7 0.5700030 0.6643347 0.07346276
9 0.01000000      8 0.5570664 0.6546181 0.07176020

Variable importance
PC1 PC3 PC2 PC5 PC4 PC6
 48  26  18   4   2   2
```

Fig. 9. Dimension Reduction result

After splitting the data, we built the SVM model. In order to tune the best hyperparameter, we did the cross validation, and found out the best gamma is 0.5 and cost is 1. The kernel we used here is the RBF kernel. The tuning performance is showing as follow:

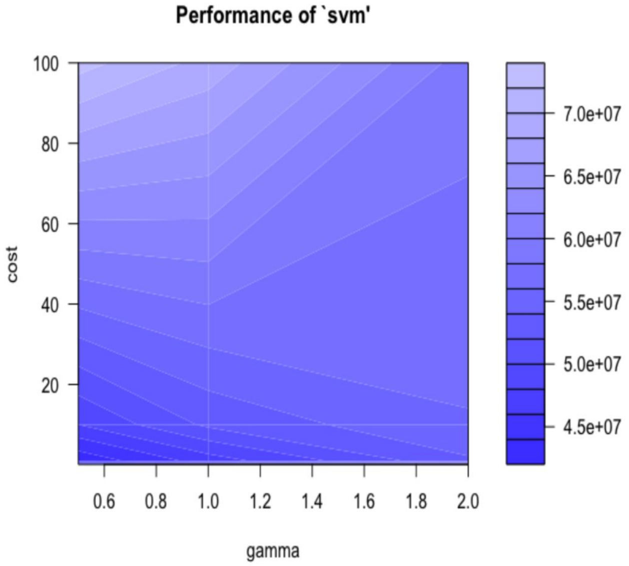


Fig. 10. SVM result

Then based on the best parameter, we built the model. Considering the large number of the earnings, we decided to use the coefficient of variation of the Root Mean Squared Error (RMSE), often seen as CV, which basically is the normalized RMSE.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y} - y_i)^2}{n}} \quad (1)$$

$$CV(RMSE) = \frac{RMSE}{\hat{y}} \quad (2)$$

Fig. 11. Rooted Mean Square Error Definition

The coefficient of RMSE of SVM is computed to be 0.1061.

The second model we built is the decision tree model. Here we tuned the parameter of max depth, we tried the depth from 1 to 10 and the tuning result is as follow:

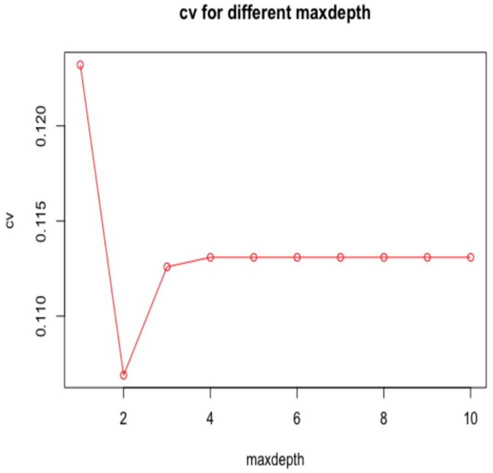


Fig. 12. Result of CV for different maxdepth

We can find out the best depth is 2, then based on that we built our decision tree model and got the CV for this model is 0.1068. The best model is as follow:

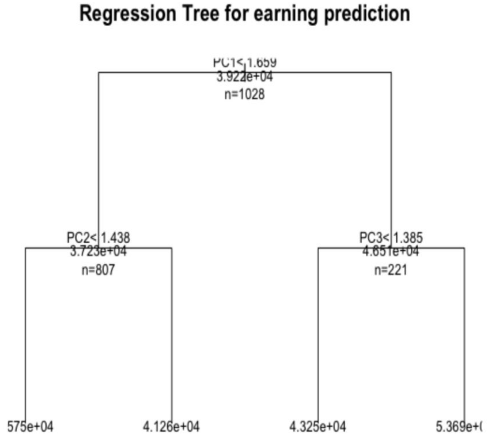


Fig. 13. Result of regression tree for earning prediction

The last model we tried is the random forest, because we want to increase our validation accuracy. Our validation set is not that large, so random forest can help us make our model more robust and generative. Here we tuned the parameter of number of the subtree. And the tuned results are as follow:

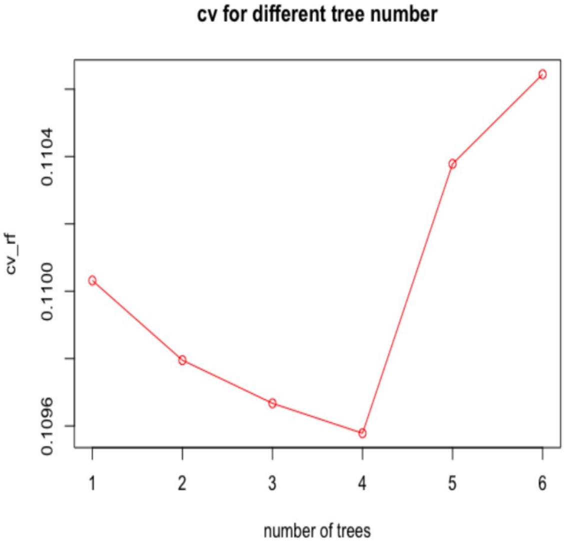


Fig. 14. Result of random forest

As we can the best number of subtree is 1100. Based on the best parameter, here we built the model and the results of CV is 0.1096.

Comparing all the model we got, we can find according to the coefficient of RMSE, the best model we think is SVM model. So we choose SVM to be our final model.

	model		
	SVM	Decision Tree	i Random Forest
CV	**0.1061**	0.1068	0.1096

Fig. 15. Comparison of all models

VII. Conclusion

In conclusion, this recommendation system is useful as it provides an unbiased opinion among the wide range of colleges. Furthermore, it tries to recommend the students to a college that is suitable to him/her and provide an earning prediction for the students to evaluate the true value of pursuing a higher education. Although, the prospective student may not depends on our recommendation system entirely, he/she will have an additional metric to rely on.

As we are doing this recommendation system, we faced some difficulties in handling big data. Since the dataset is big, we learnt that it is important to understand the data structure first and extract the useful information from the data. In addition, we also faced several challenges in extracting the data(e.g. When the data contains NULL value in some of the fields). However, because of that, we then realized the importance of data preprocessing (e.g. To remove the empty data field). Otherwise, there will be error in the result when we apply our algorithm to the inconsistent format data (e.g. data that contains empty field randomly).

Regarding the contribution, we split up the workload evenly. Each of us extracted around 2 features and generate the Bayes Factor for each features. Thereafter, we combined the data and build the User Interface together.

Acknowledgment

We would like to express our gratitude to the professor, Dr. Ching-Yung Lin and the teaching assistants in imparting us the necessary knowledge to build the final product.

References

[1] Times Higher Education. World University Rankings 2016-2017 Methodology. [online] Available at: https://www.timeshighereducation.com/world-university-rankings/methodology-world-university-rankings-2016-2017#survey-answer

[2] Huffpost.Why Your College Degree Has More Value Than You Think. [online] Available at: https://www.huffingtonpost.com/brazen-life/why-your-college-degree-h_b_3000592.html

[3] CNN. Is college worth it? Goldman Sachs says maybe not. [online] Available at: http://money.cnn.com/2015/12/09/news/economy/college-not-worth-it-goldman/index.html