Admission Prediction System Using Machine Learning

Jay Bibodi, Aasihwary Vadodaria, Anand Rawat, Jaidipkumar Patel
bibodi@csus.edu, aaishwaryvadoda@csus.edu, anandrawat@csus.edu, jaidipkumarpate@csus.edu

Abstract

We have two models as a part of Admission prediction system. The first model deals with creation of a statistical model that students can use to narrow down a set of Universities from a broad spectrum of choices. This is done using the Naïve Bayes algorithm.

The second model deals with creation of classification model which could be used by Universities for selecting suitable applicants for their programs. This is designed by establishing predefined requirement criteria. This model employs the Random Forest, Decision Tree, Naïve Bayes, SVM- Linear and SVM-Radial algorithms.

Keywords: SVM: Support Vector Machine

1 Introduction

Today, there are many students who travel to foreign countries to pursue higher education. It is necessary for the students to know what are their chances of getting an admit from such universities. Similarly, it is necessary from the university's perspective to know from the total number of applications, what will be the number of applicants who could get an admit based on certain criteria. Currently, students manually perform statistical analysis before applying to universities to find out the probable chance of getting an admit. Also, universities manually check and count the total number of applicants who could get an admit into university. These methods are slow and certainly not very consistent for students and universities to get an actual result. This method is also prone to human error and thus accounts for some inaccuracies. Since the frequency of students studying abroad has increased, there is a need to employ more efficient systems which handle the admission process accurately from both perspectives.

Our goal is to apply machine learning algorithms to admission data set. Following are the **two models**, **University Selection** and **Student Selection**. These models will not only predict and classify error and accuracy but it will also allow students and universities to pursue more simulating tasks.

University Selection model is used by the students to find the probability of the student to get an admit in the university before applying. **Student Selection** model is used by the university to analyze the results and make decision based on the classification if student would get the admission or rejection for the term student is applying for.

2 Data Set

Searching for a proper dataset was trivial in this project. Expected information from the dataset was:

- It should have necessary and sufficient columns to form a composite decision parameter based on which results can be obtained.
- It should not have a high frequency of conflicting data.
- It should be in an accessible and compatible format on which data preprocessing could be performed.
- However, such an ideal dataset was not available to the public domain on the Internet (from our previous research).

The most practical dataset found by the team members was selected from the Facebook Community called MS-in-US. The same dataset has been used to create two different datasets for constructing two different models. University Dataset for determining university decision consists of 1686 rows with 18 columns. Student Dataset is used for determining student probability of getting admit from a specific university. 10 datasets each containing 50 to 200 records of data. Original dataset has various fields like Work Experience, GRE Score, TOEFL Score, Undergrad University, Name of Student, Result, Major, etc.

2.1 Data Issues

2.1.1 Noisy Data

Specific fields that contain unfamiliar data cannot be understood and interpreted correctly by machines, such as unstructured text. For example, in a dataset, the column "Date" had many fields with improper structure. For example, some had "#" (Pound sign) instead of proper date representation.

2.1.2 Unformatted Text

Unformatted (Incompatible datatypes). Some of the data were in the string format which were supposed to be in

the integer format, a similar issue with dates. They were in different formats which had to be handled while preprocessing.

2.1.3 Inconsistent Data

Containing discrepancies (a lack of compatibility or similarity between two or more facts). Frequency of this kind of data was very high in almost all the fields where one fact was represented in multiple ways using abbreviation, code names, symbols etc. For example, university name: University of Texas, Dallas was represented in other ways like "University of Texas at Dallas", "UTD", "UT Dallas", etc. Computer Science was represented as CS, Comp Sci, Computer Sci, CSc etc.

2.1.4 Data Quality

Certain fields lack attribute values, certain attributes of interest, and contain only aggregate data. Some of the field values of the decision-making parameters were missing. Because of this some of the data had to be added. Another issue was that of aggregate data. Like the 3-different decision parameters Quantitative, Verbal and AWA (Analytical Writing Analysis) was represented as one entity under the GRE tag. Hence this composite field had to be segregated to get 3 different parameters.

2.1.5 Performance

Performance (Deteriorate without pre-processing) containing errors and outliers. Since the data was inaccurate, it was not possible to achieve the expected accuracy without removing errors and outliers. This was one of the major aspects to consider to obtain efficient results.

2.1.6 Data Skewness

"Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or data set, is symmetric if it looks the same to the left and right of the center point." [1]

"Karl Pearson coefficient of Skewness Sk = 3(mean - median) / Standard Deviation. = 3(X –Me) / S 2. The skewness of a random variable X is denoted or skew(X). It is defined as: where and are the mean and standard deviation of X." [1]

Skewness shows the inclination of the whole data set with respect to the normal distribution. In this dataset, majority of the data was of Accepted Results for a given University. Due to this the distribution was balanced to equalize the Accept and Reject fields.

3. Data Preprocessing

Following is the flowchart of the whole process.

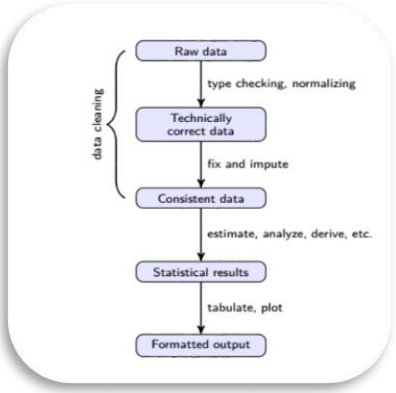


Figure 1: Data preprocessing steps

Data cleaning is performed on raw data by performing type checking and normalization. Above Data Issues are handled step by step to make sure data is consistent and compatible with the Machine Learning Algorithm

Noisy Data is handled by filtering out the unstructured text followed by changing all the values of those in proper format.
Unformatted Text: Deciding the proper format of all the fields and changing all the unformatted values into an appropriate format.
Inconsistent Data: If some data was found to be erroneous, all other values in the respective column were considered to evaluate the mean, which was then entered in place of the erroneous data.

Quality data: This was done by segregating GRE field into the 3 sub-category parameters: Quantitative, Verbal and AWA (Analytical Writing Analysis), since all these 3 sub fields are independently considered in a set of decision making parameters.

Technical Fixes: This involves handling outliers and error data and is performed solely improve the accuracy of the model. Following outliers were removed.

- Those records in which students who got less grades and those test results were accepted by the university.
- Those records in which students who got high grades and those test results were rejected.

Such kind of data create ambiguity in analysis and result.

Data Skewness has been handled by adding appropriate number of reject columns and balance it with the accepted records to get proper distribution of both accept and reject records.

After performing all these processes on the data, the dataset is finally consistent. This dataset can be used to perform the required experiments. This is followed by various tabulation and plotting schemes which can be used to obtain proper formatted information.

The University Dataset for determining decisions consists of 1686 rows with 18 columns. Student Dataset for determining student probability to get admits consists of 10 datasets each containing 50 to 200 records of data.

Result, GRE, AWA, TOEFL and **Percentage** are the columns, based on which the Student Selection model is designed.

There are 3 methods to handle missing data:

- Listwise Deletion: "Delete all data from any participant with missing values. If your sample is large enough, then you likely can drop data without substantial loss of statistical power. Be sure that the values are missing at random and that you are not inadvertently removing a class of participants." [2]

  Since our dataset was not large enough and the missing values consists of decision making parameters, deletion method was not an option.

- Recover the Values: "You can sometimes contact the participants and ask them to fill out the missing values. For in-person studies, we've found having an additional check for missing values before the participant leaves helps." [2]

  This method was practically not possible as the dataset did not have any references or ways to contact those participants.

- Educated Guessing: "It sounds arbitrary and isn't your preferred course of action, but you can often infer a missing value." [2]

  This is something which can help fix the missing value problem. But rather than going for an arbitrary guess, we chose the mean as the substitution method for missing values. This ensured that the guessed values are not outliers but, fit well within the domain.

We ignored the record where percentage was not present. Here Listwise Deletion method is used. The number of missing values in percentage were very few compared to the total number of records in the whole dataset. This method is comparatively feasible and appropriate.
Changing categorical data to numeric value. All operations and functions were done on numeric value, so all categorical values must be converted into proper numeric form. e.g. Results.
Feature Scaling was done on all the columns except the Results field as it only contains Accept or Reject values. Normalization was performed on required fields so that various columns could be compared at the same base.
Following are some of the Original Dataset Representation which can help to understand the nature of it rather than going through the whole excel sheet data which is time consuming.

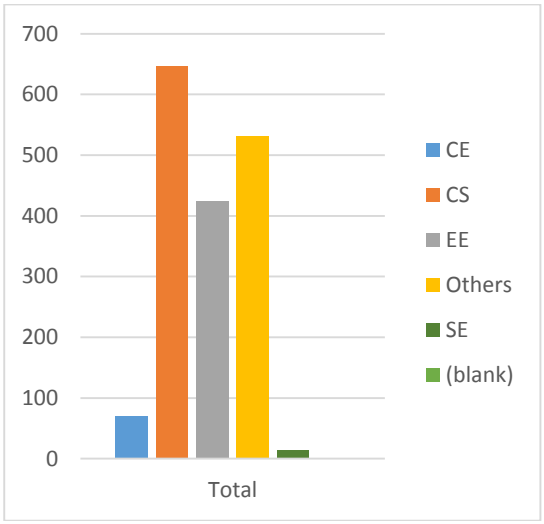


Figure 2: Distribution of major

The above graph is the representation of various majors and their distribution frequency. On X-axis is the major and on Y-axis is its corresponding number of records. Here majority of the data records are of Computer Science and hence that is taken into consideration for both the models.

Due to limited amount of data available for other majors, it is very difficult to maintain good accuracy. 'Others' contain different majors other than the once mentioned here.

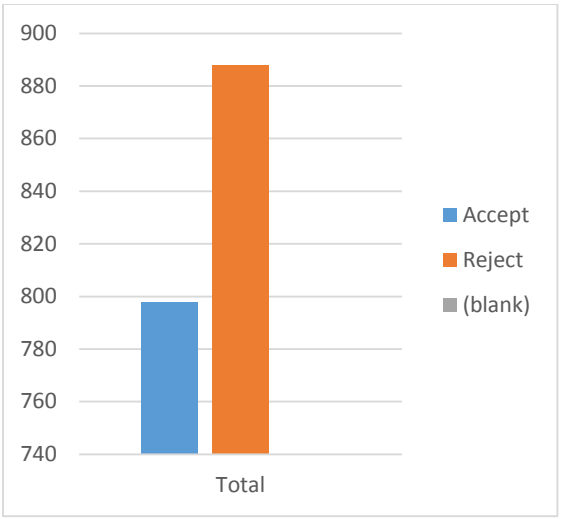


Figure 3: After pre-processing distribution of result

Since there was an imbalance in the number of Accept and Reject records, it was modified and new data was added to balance this issue. After doing this, since the model required more information of the Reject fields than the Accept fields, dataset was modified again by increasing the number of Reject fields by around 100 more than that of Accept.

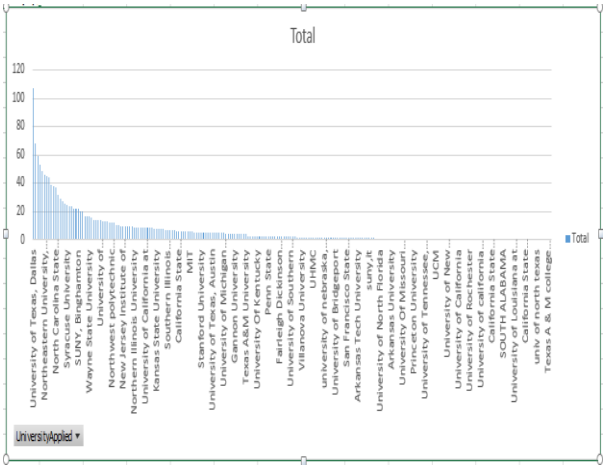


Figure 4: Frequency distribution of University

Frequency distribution of the dataset grouped by Individual University is shown above. On X-axis, is the list of different universities. On Y-axis, is the number of records available for each university. Here the data of University of Texas, Dallas has the highest number of records. As number of records for other university is very less, we are limiting the scope of the Student Selection model to this university. The same process can be done on other universities to obtain similar results.
In Student selection model, 10 datasets of specific universities were created to obtain the probability of a student against each of these universities.

4. Model Development

4.1 Preliminaries:

Machine learning classification technique is a supervised learning that is designed to infer class labels from a well-labeled trained set having input features associated with the class labels. [3]
After cleaning the data as mentioned in the prior section, our two models can be designed as

- "University Selection" Model – Classification problem with apriori probability output.
- "Student Selection" Model – Classification using supervised learning.

For the "University Selection" Model, we use the Naïve Bayes classifier. Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. [4]

Naive Bayes has been a popular (baseline) method for text categorization, i.e. the problem of judging documents, belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. [4] Maximum-likelihood training can be done by evaluating a closed-form expression like

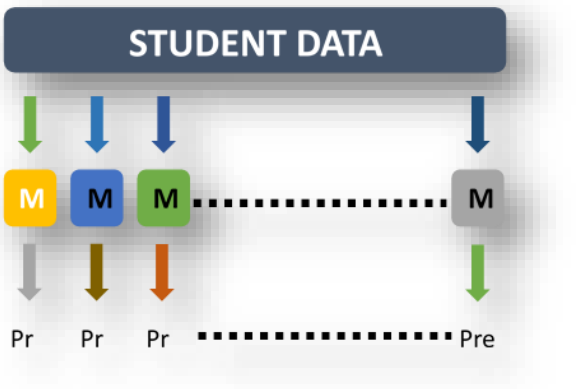
$$posterior = \frac{prior \times likelihood}{evidence}$$

Equation 1: Naive Bayes *[4]*

Which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers such as decision trees and SVM.

For the second model, "Student Selection" we worked with a variety of models, namely Naïve Bayes, SVM, Decision Tree and Random Forest.

Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of



the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. [5]

The decision tree algorithm is a machine learning classification mechanism, where patterns of input features are analyzed to create a predictive model. A decision tree consists of non-leaf nodes representing tests of features, branches between nodes representing the outcomes of the tests, and leaf nodes holding the class labels. [3]

Constructing the most optimal and accurate decision tree is usually NP-hard on a given training set [3]. To construct a decision tree model, most of the practical algorithms use a greedy approach using heuristics such as information gain. Using these algorithms, the training data is recursively partitioned into smaller subsets. When partitioning the dataset, the feature with the highest splitting criterion such as information gain is chosen as the splitting feature. This feature minimizes the information needed to classify the data in the resulting partitions and reflects the least randomness in these partitions.

The Random forests method consists of multiple decision trees that are constructed by randomly chosen features with a predefined number of features. The random features classify a label by voting, a plurality decision from individual decision trees. Because of the law of large numbers, the Random forests method is less prone to generalization error (overfit) as randomness are added with more trees. In addition, the generalization error converges to a limited value. It is due to this property of Random Forests, we achieved the accuracy of ≈ 90%

4.2 University Selection

Since the main aim of the model is to find the probability of admission of a student given his scores and other attributes, we choose Naïve Bayes Classifier. This classifier, as mentioned before estimates the classification on the basis of the probability. This fits right into our requirement. We started the pre-processing by extracting top 10 (in terms of a number of records) university data from the original dataset $\boldsymbol{D}$ into 10 separate datasets. Each dataset $\boldsymbol{d_i}$ is used to train a model $\boldsymbol{M_i}$.

Figure 5: University Selection System

The flow chart above gives a basic idea regarding the functioning of the system.

After all the models are generated, any new students information is evaluated against all the models and their corresponding prediction for acceptance $\boldsymbol{P_i}$ is collected into a pool of predictions. This pool is then sorted in descending order to provide the top 5 probable universities. Given below is one such example.

Table 1: Probability pool

University	Probability
MTU_pred	0.96610169
clemson_pred	0.90909091
NE_Boston_pred	0.82608696
ASU_pred	0.82352941
IITchicago_pred	0.80000000
RIT_pred	0.76923077
UTD_pred	0.21296296
UTA_pred	0.18867925
UNC_pred	0.18421053
U_southern_cal_pred	0.08163265

Table 2: Sample student data

GRE	AWA	TOEFL	IELTS	Percentage
309	3.5	90	N/A	85

As per the output, the student in Table 2 has the highest probability of getting into Michigan Technological University with the probability of 0.96. Followed by Clemson University with 0.90 probability. Using this output the student can decide which universities to apply for.

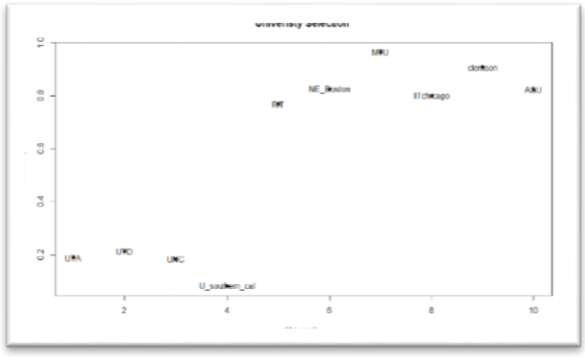


Figure 6: Unsorted Probability Output

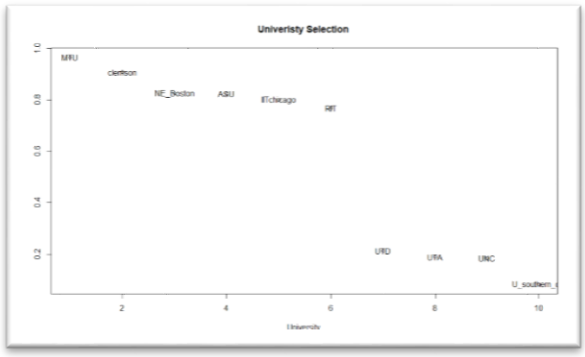
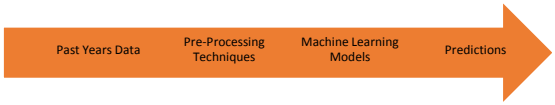


Figure 7: Sorted probability output

4.3 Student Selection

The main aim of this system is to classify new applications based on previous years data of the students who got admits or rejects in a particular university. Due to the constraints of the data size, we choose to build this system for University of Texas, Dallas.
Given below are the steps for developing this system:



After pre-processing the data as mentioned in the earlier section, we train different supervised classification models to classify applications into '***Accept***' or '***Reject***'. The different models used are ***Naïve Bayes, SVM – Linear & Kernel, Decision Tree, and Random Forest***. Each of these models was used to test the set of new applicants along with result to derive the accuracy.

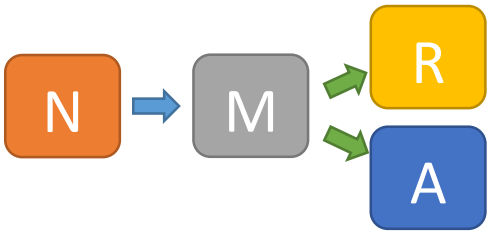


Figure 8: Student selection system

N: Represents the new applicants applying to the university

M: Different Models as mentioned

R: Class Reject

A: Class Accept

We saw the highest error rate in ***SVM – Kernel*** of ***13.06%***. Followed by ***SVM – Linear*** of ***12.56%***. ***Naïve Bayes*** produced an error rate of ***12.06%***. We received the best results in ***Decision Tree*** and ***Random Forests***. Given below is the decision tree modeled as per our dataset.

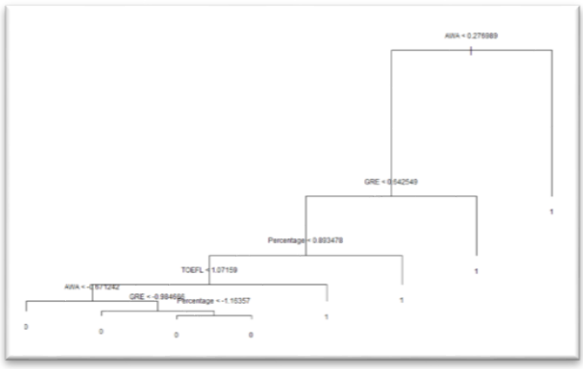


Figure 9: Decision tree

Here ***1*** represents '***Accept***' and ***0*** represents '***Reject***'.

As per the tree, the most important criteria checked is ***Analytical Writing Assessment*** from ***GRE***. If any student scores above a certain threshold, he/she is accepted. If not, the ***GRE*** is checked and the candidate is accepted if he/she scores above a set threshold. The third important criterion is ***Undergrad Percentage***. Any candidate with a percentage higher than ***89%*** is accepted. The least weighted criterion is ***TOEFL***, according to which if any student is below the median then they'll be rejected. We are able to achieve an accuracy of ***≈ 89%*** using Decision Tree.

For Random Forest, first, we had to decide the number of trees to generate for the forest. We used ***Out-Of-Bag (OOB) Error***. [6]

'*Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree.*

Put each case left out in the construction of the kth tree down the kth tree to get a classification. In this way, a test set classification is obtained for each case in about one-third of the trees. At the end of the run, take j to be the class that got most of the votes every time case n was oob. The proportion of times that j is not equal to the true class of n averaged over all cases is the oob error estimate. This has proven to be unbiased in many tests.' [6]

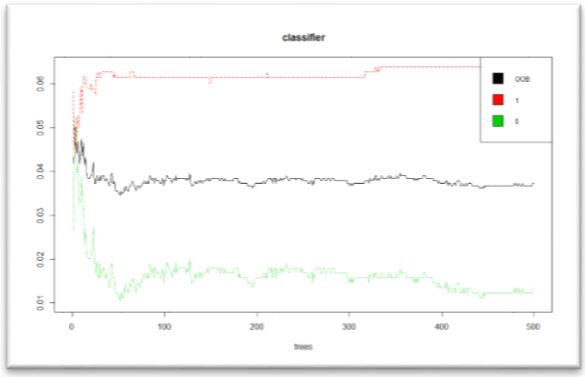


Figure 10: Error rate vs number of trees graph

In the above graph, Green represents '***Reject***' error rate, Red represents '***Accept***' error rate and Black represents '***OOB***' error rate. We can see that optimal number lies between 60 and 100. For our model, we used 70 trees.

Using this Random Forest we achieved an accuracy of ***≈ 90%***

5. Future Enhancements

Creating the model with additional parameters such as Work Experience, Technical Papers Written, and rating the Content of Letters of Recommendation etc. can make it more flexible to the Universities admission requirements. Hence by generalizing the decision-making parameters, this system can be used for any admission prediction process by taking into consideration all desired criteria.

Creating a model based on the graph of admitted vs enrolled students of previous years to predict the increase or decrease in cutoff scores among applicants which will be useful from the university perspective in the long run to analyze applicants who apply for each term.

Comparing different universities based on applied vs admitted data so that students before applying to any university could measure variations of the admits and rejects of the university.

6. Learning

Give below are our learnings for this project

- Data preprocessing is vital to the accuracy of the model.
- Choosing appropriate machine learning techniques and algorithms to model the system
- Graphical representation of the data provides useful insights and can lead to better models.
- Defining scope with respect to the dataset

Appendix

Find all the Support material using below link

http://athena.ecs.csus.edu/~pateljd/support.html

1. Raw Data (Fall_2014.csv)
2. University selection Model
   - Input data (stu_csv.rar)
   - Source Code (Student.R)
   - Output (stu-output.rar)
3. Student Selection Mode
   - Input Data (uni_csv.rar)
   - Source Code (University.R)
   - Output (stu-output.rar)

References

[1] "Skewness," [Online]. Available: https://www.uky.edu/Centers/HIV/cjt765/9.Skewness%20and%20Kurtosis.pdf. [Accessed 17 05 2017].

[2] J. Sauro, "MeasuringU: 7 Ways to Handle Missing Data," MeasuringU, [Online]. Available: https://measuringu.com/handle-missing-data/. [Accessed 17 05 2017].

[3] J. R. Quinlan, "Induction of Decision Trees," Mach Learn, 1986.

[4] Wikipedia, "Naive Bayes Classifier," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier. [Accessed 16 05 2017].

[5] Wikipedia, "Support Vector Machine," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine. [Accessed 16 07 2017].

[6] L. B. a. A. Cutler, "Random forests - classification description," Salford Systems, [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr. [Accessed 17 05 2017].

[7] P. C. a. A. Silva, "USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE".

[8] H. W. a. Z. Y. Rensong Dong, "The module of prediction of College Entrance Examination aspiration".

[9] D. T. E. S. L. R. a. A. P. William Eberle, "Using Machine Learning and Predictive Modeling to Assess Admission Policies and Standards".

[10] H. M. Havan Agrawal, "Student Performance Prediction using Machine Learning," IEEE.