Predict Whether One Should Apply for Graduate School through Classification in Machine Learning

CS 178 – Winter 2018

Hu Haotian
Liu Haosong
Qiu Shiyu
Tan Conghuai
Yin Tingjue

I. Problem Statement

It is expensive to apply for graduate schools in the U.S. because undergraduates need to spend plenty of time and money on examinations, recommendation letters and applications. If the students get rejected, they will end up with nothing after making a lot of efforts on it. Therefore, to save time and money, our project aims to give suggestions on whether an Indian undergraduate should apply for graduate schools in the U.S. based on his or her GRE score, TOEFL score, university rating, SOP, LOR, CGPA and research through classifying the chance of acceptance as low, medium or high.

We trained the data of Indian undergraduates who applied for U.S. graduate schools and fitted and compared several models of classification in order to find a suitable classifier to make the prediction. Then, our project can predict and generate one of the three outcomes to the student: apply for graduate schools because of high chances of admission, try to apply for graduate schools but may be rejected or not suggest applying because of low chances of admission.

II. Dataset

The dataset is downloaded from Kaggle.com and created by Mohan S Acharya. It has a shape of 500 rows and 9 columns. That is, it has 500 data and 9 features: “Serial No.”, “GRE Score”, “TOEFL Score”, “University Rating”, “SOP”, “LOR”, “CGPA”, “Research” and “Chance of Admit”.

We first summarized the data and checked the missing data. Luckily, the dataset is pretty clean except “Serial No.” which is not an influential feature in the dataset so we removed it. Then, to visually analyze the data, we draw a pair plot (figure 1). The histograms of the features look reasonable and the relations between some features also make sense. Generally, the dataset is unbiased and representative.

We want to emphasize the feature “chance of admission” which is a number ranging from 0 to 1 meaning the percentage of acceptance. It is what we will predict so it is removed from the features and becomes the Y value. Also, our project aims to directly suggest the student whether one should apply for graduate schools instead of telling a plain number so we classify “chance of admission” into 3 classes: low, medium and high chance using the threshold of 0.66 and 0.8, because this yields 3 classes with the same number of data points.

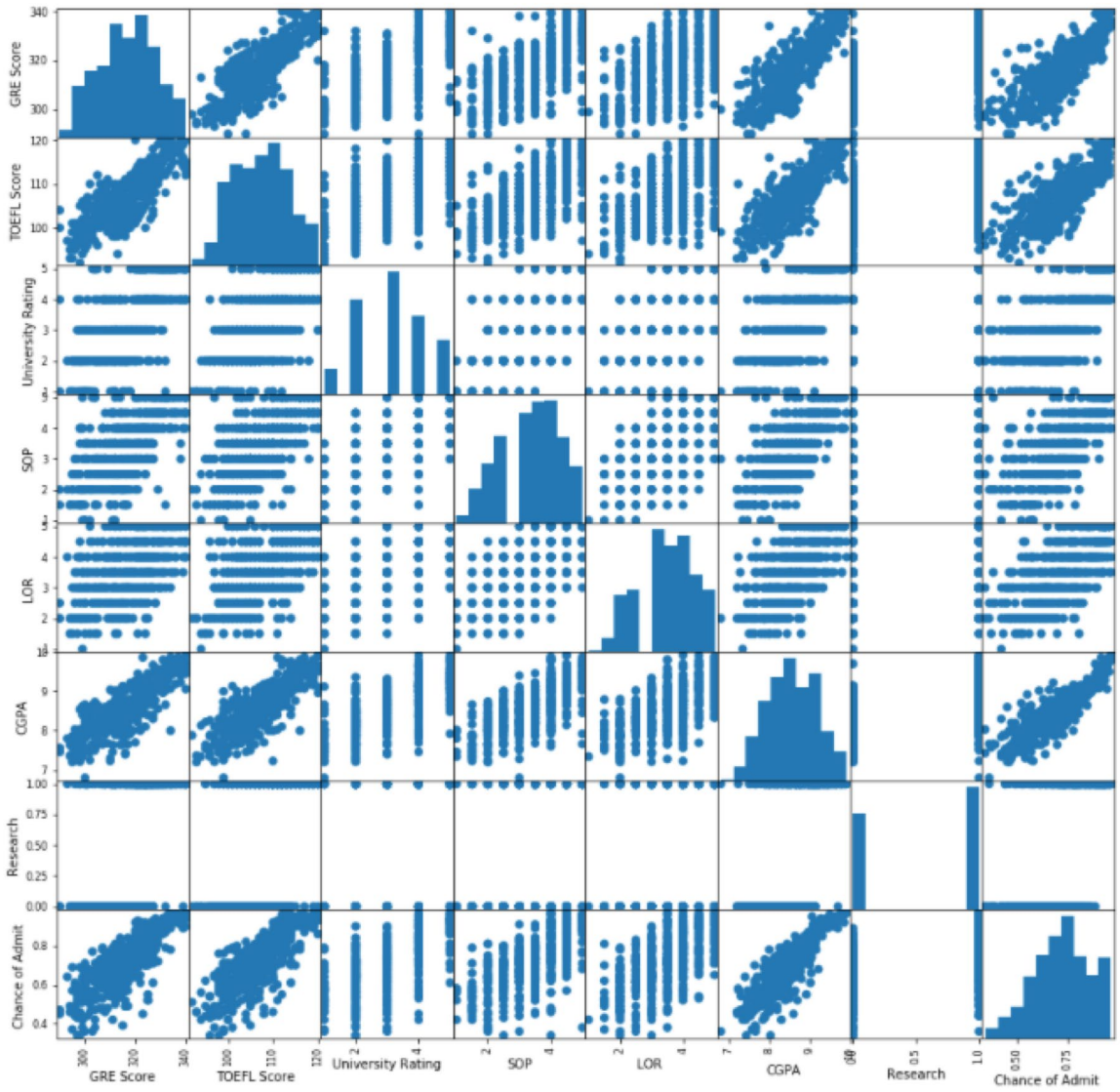


Figure 1 shows a pairplot of the dataset.

III. Literature Review

It is quite popular for both college admission offices and students who apply for colleges to use predictive computational models to forecast one's academic performance. College admission committees often attempt to predict an applicant's academic success as a means of assisting the admission decision by selecting candidates' performance under the professional curriculum[siu]. For example [Dun], the study shows undergraduate grade point averages (UGPAs) and Medical College Admission Test (MCAT) total scores are strong predictors of academic performance in medical school through graduation using logistic regression. Another study [Mura] uses the random forest machine learning technique to develop a binary classification model that has an overall accuracy of 77% for candidates with high or low academic performance. It can be used as preliminary filters during the admission process.

For students, they can use similar models that forecast the academic performance to predict their college admission chances. Methodologies such as simple and multivariate linear regression, as well as logistic regression, are frequently mentioned by institutes[use].

However, in Chang's study [ ], he "compared several classification approaches to predict admissions for a university in the United States. Specifically, he compared logistic regression with other data mining techniques, concluding that the data mining approaches achieved superior performance compared to logistic regression".

Except using the models, many researchers studied various factors that can lead to a higher chance of being admitted into elite colleges, such as being a male athlete [esp], taking nine or more AP exams [Esp], higher SAT scores [] and being top 10 students in high schools [new].

IV. Decomposition of Work

All five of our group members worked actively, diligently and cooperatively. We met with each other and worked collaboratively four times in total. For the first time of the meeting, Conghuai and Haosong shared several useful links about the possible project topics. After reading all the information, we discussed and decided our topic along with a general framework of the project. It was the first checkpoint.

We elaborated the project by defining the problem, coding and choosing models for the second time of working together. The dataset we found didn't have a complete and detailed description so we searched online to see whether the creator of the data left any useful comments or discussions. Then, Haotian cleaned and analyzed the data, such as drawing several plots to learn the relations between features. Next, our project had two possible directions: predicting a percentage of being accepted using regressions or predicting whether one would be accepted using classifications. Our possible models include linear regression, logistic regression, KNN, Gradient boosting, decision tree and random forest. We discussed many possible ways for the whole afternoon. For instance, we tried to use both regressions and classifications and choose the best one but we realized that they can't be compared. Or, we wanted to find the best models for the two ways separately. As a result, five of us wrote down both regressors and classifiers for the possible models.

For the third time of the meeting, we decided the methods which were a milestone: we would use classifications -- logistic regression, KNN, random forest and gradient boosting. Also, we decided to classify the data into three classes: low, medium and high chance of admit. Tingjue clarified the following steps of working. Each of Haotian, Haosong, Shiyu, Conghuai wrote one model in the Python script and wrote down the corresponding experience and explanations in the models and conclusion parts of the report. Tingjue wrote the remaining parts of the report. Besides, Haotian combined all the codes together.

Lastly, we drew the poster together during the fourth time of the meeting.

V. Models

1. Logistic Regression

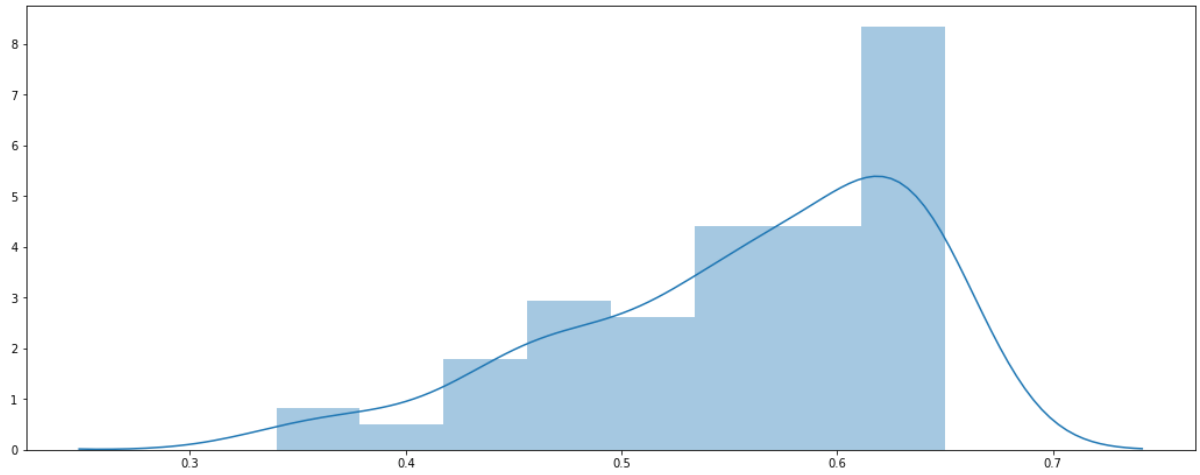
Concept:

From basic, logistic regression is very similar to linear regression. Instead of performing gradient descent on a multinomial linear function, logistic regression performs gradient descent on an S-shaped sigmoid function, so that finding the best degree of multinomial would not be necessary. We know that logistic regression works best when the result is labeled dichotomously, but we still wanted to give it a try.

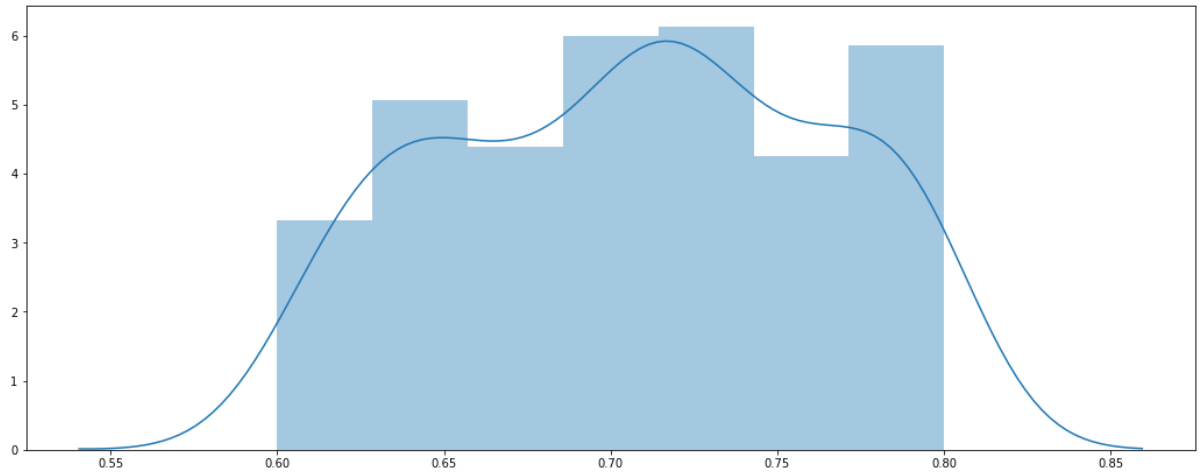
Train:

First threshold the data so that we have 3 classes of the same number of data points. After drawing the histogram and analyzing the distribution of the data, we found that 0.66 and 0.80 can do the job.

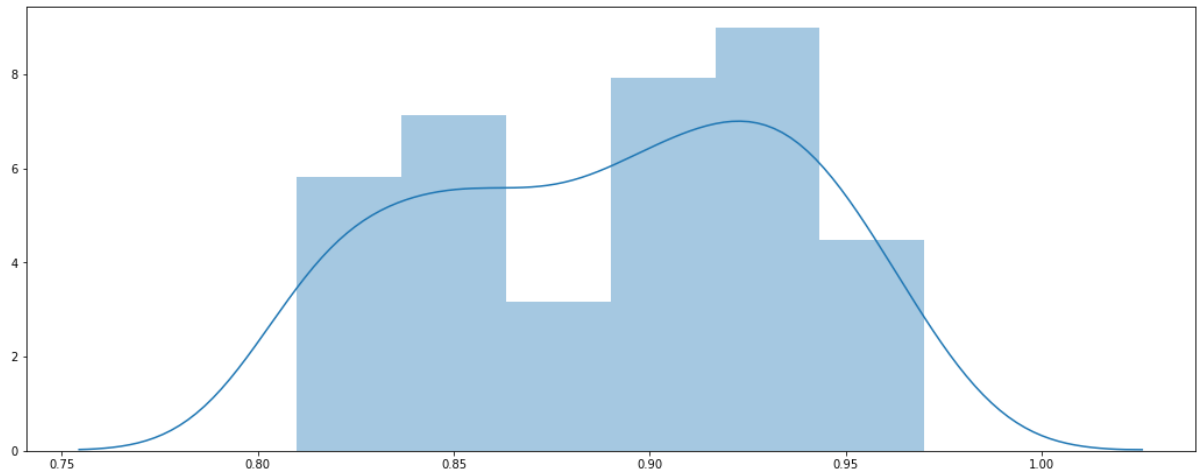
Class 1(y < 0.66):



Class 2(0.80>= y >= 0.66):



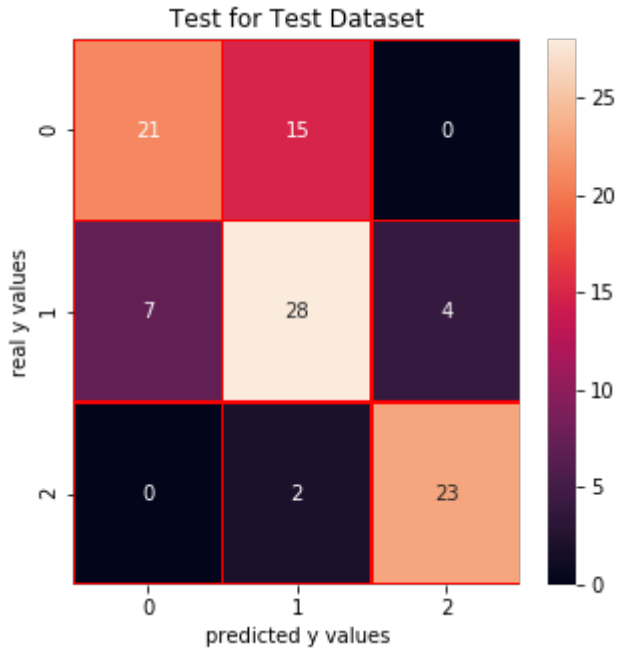
Class 3(y > 0.80):



Result:

We fit the model using LogisticRegression from the sklearn.linear_model library and got the following result:

The mean accuracy of Logistic Regression model is: 0.72



2. K-Nearest Neighbor Classifier

Concept:

K Nearest Neighbors is a non-parametric, lazy machine learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. The 'non-parametric' characteristic implies that the algorithm does not make any assumptions and the model structure is only determined from the data we have. KNN is also an 'lazy' algorithm but it does not mean KNN is doing nothing, 'lazy' means KNN does not use training data points to do any generalization. Such two characteristics reveal that KNN is an algorithm calculating data based on the 'real world', so KNN could and probably should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution data.[Bronshtein, Adi. "A Quick Introduction to K-Nearest Neighbors Algorithm." *Medium*, Medium, 11 Apr. 2017, medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.]

K Nearest Neighbors algorithm has clear advantages and disadvantages. The advantages are KNN does no assumption about data, it is a simple algorithm, it has high accuracy(relatively), it is robust to noisy training data and it is effective if the training data is large. The disadvantages are it needs to determine the value of parameter K(the number of nearest neighbors), its distance based learning is not clear which type of distance to use and which attribute to use to produce the best results, and the computation cost is quite high because we need to compute the distance of each query instance to all training samples.[Bronshtein, Adi. "A Quick Introduction to K-Nearest Neighbors Algorithm." *Medium*, Medium, 11 Apr. 2017, medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.],[Teknomo, Kardi. "Strength and Weakness of K-Nearest Neighbor Algorithm." *K Nearest Neighbors Tutorial: Strength and Weakness*, people.revoledu.com/kardi/tutorial/KNN/Strength%20and%20Weakness.htm.]

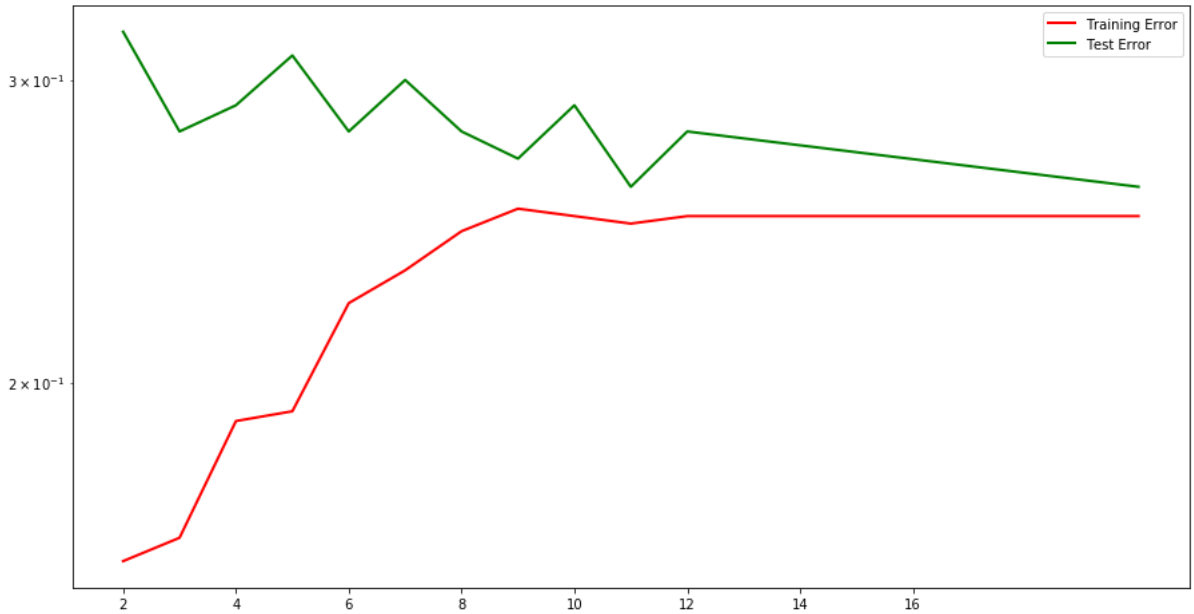
KNN Algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point. We choose KNN classification—the output is a class membership (predicts a class—a discrete value). An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.[Bronshtein, Adi. "A Quick Introduction to K-Nearest Neighbors Algorithm." *Medium*, Medium, 11 Apr. 2017, medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.]

Train:

In our data, we have 7 features which are GRE scores, TOEFL scores, University Ratings, Statement of Purpose, Strength of Recommendation letter, Undergraduate GPAs, and Research Experience, relate to the Chance of Admit.

First, we split 'Chance of admit into 3 classes' which are [0]: Lower than 66% , [1]: Bigger than 66% but lower than 80%, and [2]: Bigger than 80%. Then I split the data into training sets and testing sets by ratio of the amount in 4:1.

Then, I need to determine the best k value to use in the KNN model. Using KNeighborsClassifier built in [sklearn] library to test the training mean square error and testing mean square error difference. The graph below shows the relationship between training mean square error and testing mean square error.



From the graph, we could see that the best value of k is 11.

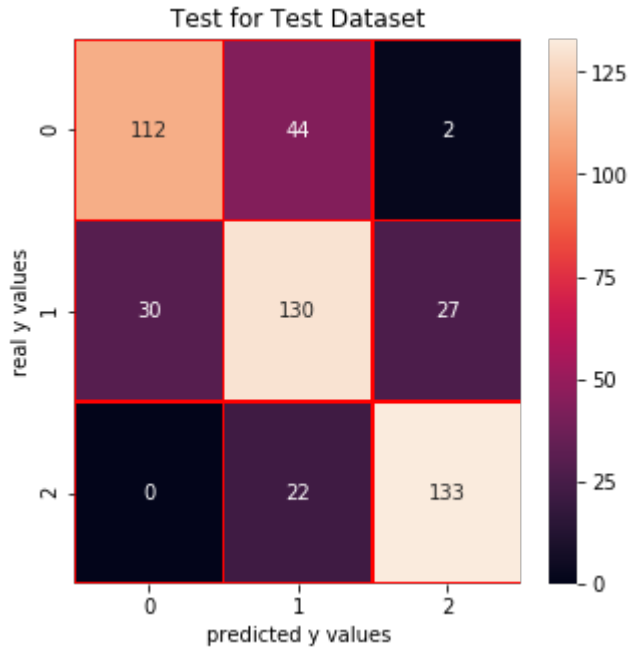
Result:

Using 11 as the value of k, again, using KNeighborsClassifier in [sklearn], we could see:

1. Mean Accuracy:

```
The score of Knn is :  0.74
```

2. The Confusion Matrix:



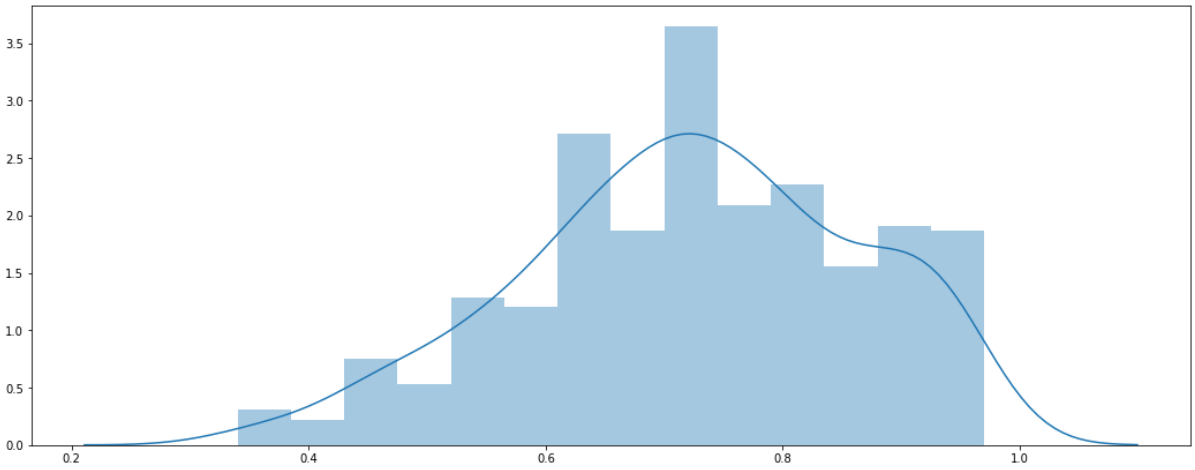
3. Random Forest

Concept:

Random Forest operates by constructing multiple decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. Random forests deduct the risk of overfitting to the training set.

Train:

First, process the data. Delete the first column which is the serial number of different datum. Separate the columns into X and Y; X is the first six columns which are six features, and Y is the last column which is the target value, probability from 0 to 1. Then separate the Y set into 3 classes -- high, medium and low, according to two thresholds 0.8 and 0.66. The reason for choosing these two thresholds is because of the histogram graph for the Y set. According to the Y set histogram, 0.8 and 0.66 can set the data of different classes to a proper amount. Select 20% of X and 20% of Y as the test X set and the test Y set, and the left data as the training set.



Second, build a random forest classifier using the default parameter values. With n_estimators = 100 and max_depth = 7, this classifier can predict for the test data set with a score of 0.82 out of 1.

```
score of classifier with default parameter values: 0.81
```

Third, choose the optimal value for the parameter max_depth. Given a list of different max_depth values, build a classifier with each different element from the list, and collect the mse for training data and test data for different max_depth value. Plot training mse set and test mse set, and find the lowest point among the test mse. According to figure (1), the corresponding max_depth value to the lowest test mse value is the optimal choice for max_depth, which is 8. Set max_depth to 8 and use the same algorithm to find the optimal value for the parameter n_estimators. According to the figure (2), 90 is the optimal choice for n_estimators.

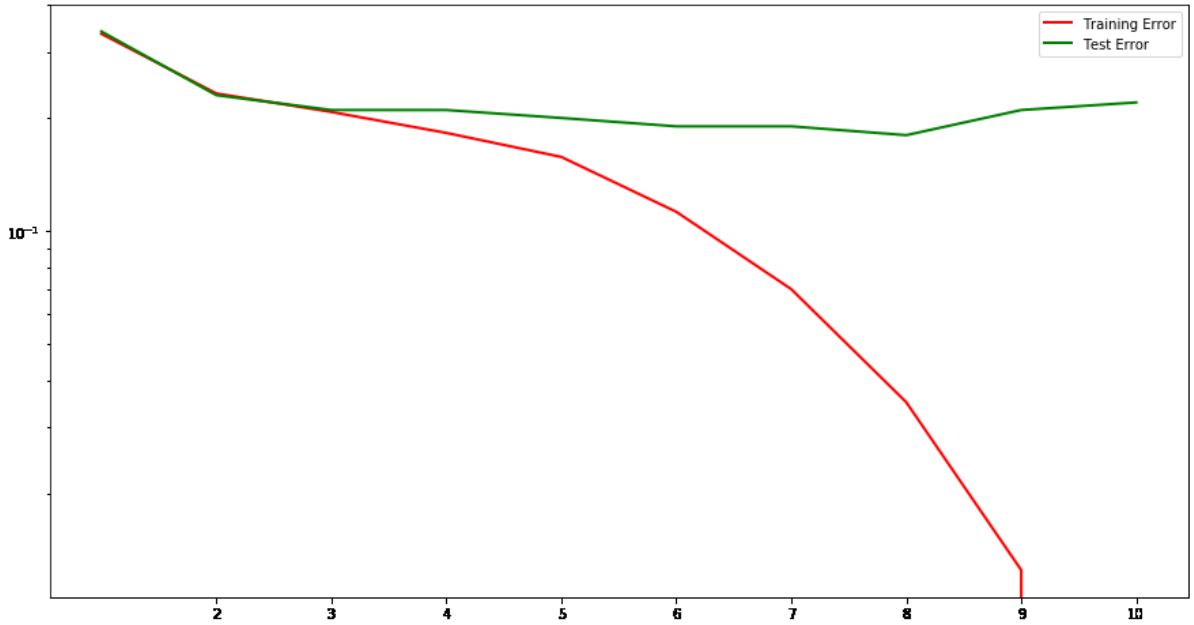


figure (1)

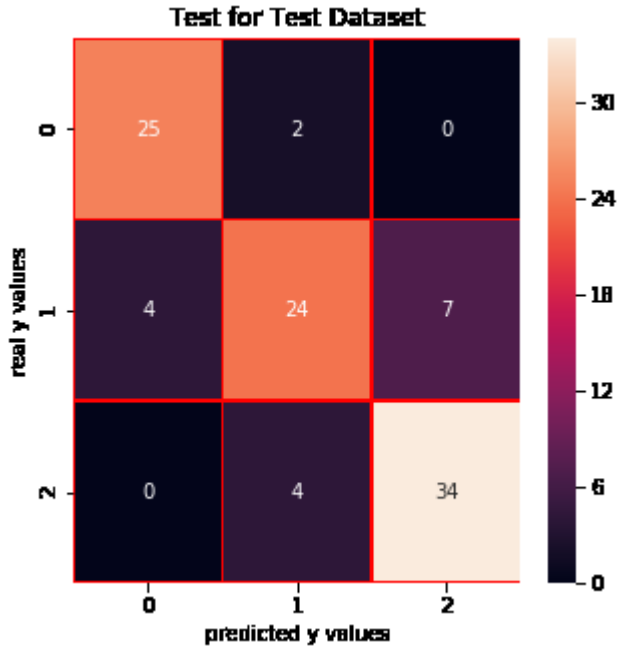
figure (2)

Result:

The score of predicting test data using the classifier built with optimal parameter values is 0.83 out of 1.

```
score of classifier with optimal parameter values: 0.83
```

As shown in the confusion matrix below, most of the predicted results overlap with the test Y set. The lighter the color, the more data fall in the cell.



4. **Gradient Boosting Classifier:**

Concept:

Gradient Boosting is a lot of models come together to make the prediction. In this case, the decision tree is chosen as the prediction model and I use a bunch of decision trees to work together to make Gradient Boosting come true.

Train:

I first split the data into 3 classes (based on Chance of Admit):

1. Lower than 66%
2. Bigger than 66% but lower than 80%
3. Bigger than 80%

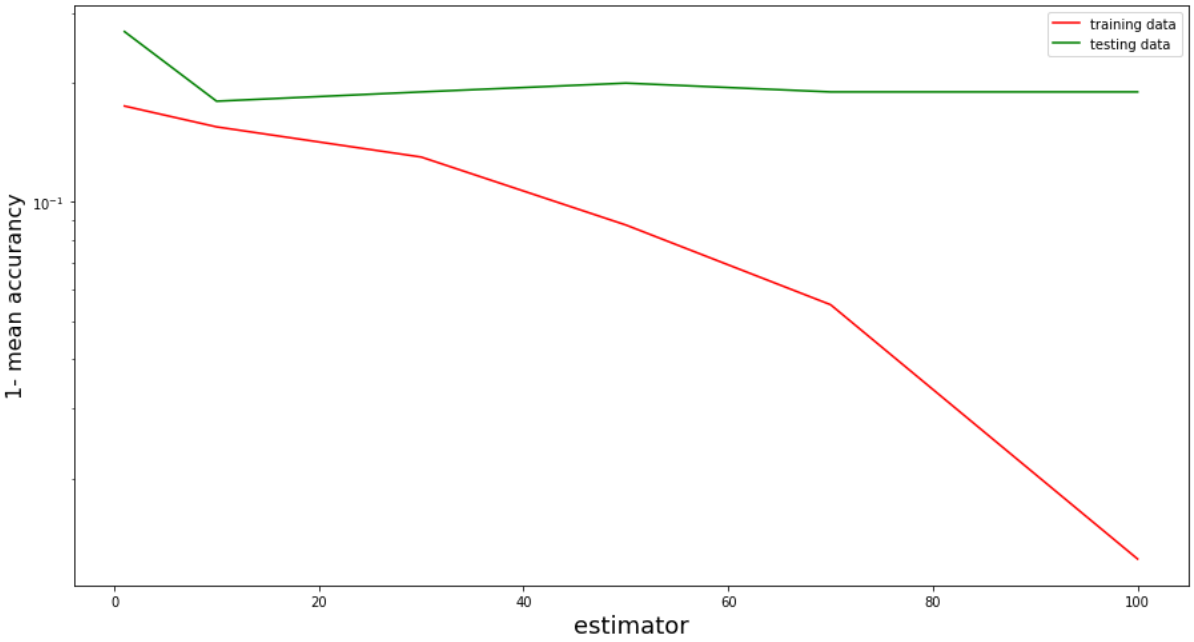
Then I split the data into training sets and testing sets by a ratio of the amount in 1:4

Based on my research on the Gradient Boosting, there are two important constraints to improve the accuracy of Gradient Boosting Classifier:

1. Number of decision trees to choose to use in Gradient Boosting Classifier
2. The depth of each decision tree

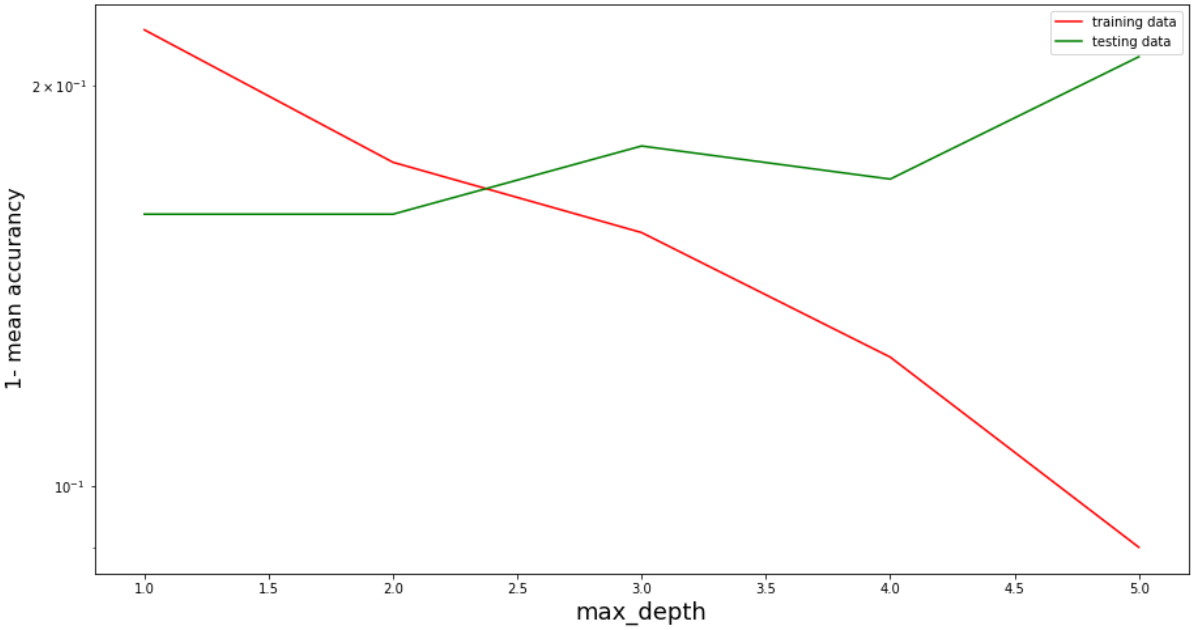
1. Choose the number of decision trees (estimators) to be **10**:

The graph below shows the relationship between the estimator and (1- mean accuracy). As the graph shows, when estimator equals to **10**, the (1-mean accuracy) is the lowest. It means the mean accuracy is the highest.



2. Choose the depth of each decision tree to be **2**:

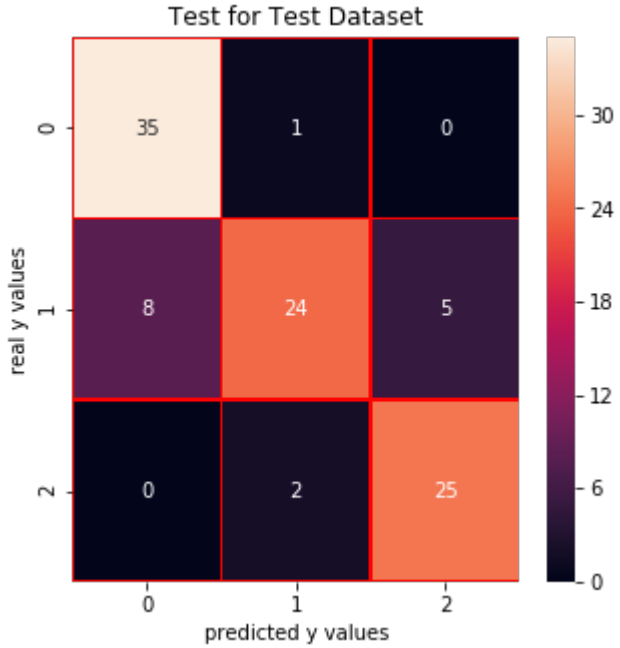
The graph below shows the relationship between max depth and (1- mean accuracy). As the graph shows, when max depth equals to **2**, the (1-mean accuracy) is the lowest. It means the mean accuracy is the highest.



Result:

Used two method to show the results of best Gradient Boosting (choosing n_estimators = 10, max_depth = 2):

1. Confusion Matrix (visualization):



2. Mean Accuracy:

```
The score for is GradientBoostingClassifier: 0.84
```

VI. Conclusions

（通过results写这个model的优点和缺点， Is there an analysis of the achieved results/contributions and why it worked/not worked? And how it can be improved?）

1. 每个人解释下自己model的优点和缺点（比如容易overfit)
2. 通过以上的信息，我们决定用Gradient Boosting， 因为:……
3. 总结整个project可以在哪里提升

1. Logistic Regression:
   a. Advantages:
      i. Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary).
      ii. Compare to linear regression, we don't need to perform the polynomial transformation to find the ideal degree to fit the data.
   b. Disadvantages:
      i. However, in our case, the dependent variable is not binary, because we decided to separate the admission rate into three classes.
   c. Possible improvement:
      i. Separate the training phase into 3 part:
         1. Class 0 vs. Class 1
         2. Class 1 vs. Class 2
         3. Class 2 vs. Class 0

ii. Fit the data this way to get a better accuracy result because of the nature of Logistic Regression.

2.
3. Random Forest:
    a. Advantages:
        i. The Random Forest Model can avoid overfitting because there are enough decision trees.
        ii. Comparing to logistic regression, random forest is more accurate
    b. disadvantages:
        i. For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data. Methods such as partial permutations were used to solve the problem.
        ii. If the data contain groups of correlated features of similar relevance for the output, then smaller groups are favored over larger groups .
4. Gradient Boosting:
    a. Advantage:
        1. Boosting is one of the method of Ensembles. Ensembles is one of the best method to help to reduce the variance and bias efficiently. In this way, it bits the Logistic Regression and K-Nearest Neighbor Classifier by having a better mean accuracy/score and confusion matrix.
        2. It takes less time and iteration to fit the data into the model because Boosting is built by bunch of same predictor (example: decision tree) and each predictor is sequentially related with each other. Each predictor is fitted and have different weigh based on the mistakes committed by previous model. Therefore, it bits the Random Forest by having less iteration time to fit the data.
    b. Disadvantage:
        1. The Gradient Boosting is easy to overfit and therefore need to find a good threshold to stop the procedure of fitting and split the testing and training data carefully.

Choose the Best Model: (Gradient Boosting Classifier)

VII. Bibliography

Espenshade, T. J., Chung, C. Y., & Walling, J. L. (2004). Admission Preferences for Minority Students, Athletes, and Legacies at Elite Universities*. *Social Science Quarterly,85*(5), 1422-1446. doi:10.1111/j.0038-4941.2004.00284.x

Espenshade, T., Lauren E. Hale, & Chung, C. (2005). The Frog Pond Revisited: High School Academic Context, Class Rank, and Elite College Admission. *Sociology of Education, 78*(4), 269-293. Retrieved from http://www.jstor.org/stable/4150499

Dunleavy DM, Kroopnick MH, Dowd KW, Searcy CA, Zhao X. The predictive validity of the MCAT exam in relation to academic performance through medical school: a national cohort study of 2001- 2004 matriculants. Acad Med. 2013;88(5):666-671.

L. Chang, Applying data mining to predict college admissions yield: A case study, New Directions For Institutional Research 131 (2006), 53–68.

Muratov E, Lewis M, Fourches D, Tropsha A, Cox WC. Computer-assisted decision support for student admissions based on their predicted academic performance. Am J Pharm Educ. 2017;81(3) Article 46.

Newport, C. (2010). *How to be a high school superstar: A revolutionary plan to get into college by standing out (without burning out)*. New York: Broadway Books.

Mohan S Acharya, Asfia Armaan, Aneeta S Antony : A Comparison of Regression Models for Prediction of Graduate Admissions, IEEE International Conference on Computational Intelligence in Data Science 2019

Siu E, Reiter HI. Overview: what's worked and what hasn't as a guide towards predictive admissions tool development. Adv Health Sci Educ Theory Pract. 2009;14(5):759775.

Use of Predictive Validity Studies to Inform Admission Practices. (n.d.). Retrieved from https://www.nacacnet.org/news--publications/Research/report-validity-studies/

Jason Brownlee: A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning, September 9, 2016 https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

Prince Grover: Gradient Boosting from scratch, Dec 8, 2017, https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

Learn. (n.d.). Retrieved from https://scikit-learn.org/stable/

Raphael Couronné, & Philipp Probst, A. (2018, July 17). Random forest versus logistic regression: A large-scale benchmark experiment. Retrieved from https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2264-5

Ravanshad, A., & Ravanshad, A. (2018, April 27). Gradient Boosting vs Random Forest. Retrieved from https://medium.com/@aravanshad/gradient-boosting-versus-random-forest-cfa3fa8f0d80