Graduate School Admission Chances Prediction

Haozhi Hu
A92102421
Department of CSE, UCSD
hah045@ucsd.edu

ABSTRACT

Universities usually publish detailed reports on undergraduate admission and enrollment; however, the same does not hold for graduate schools. Most graduate programs keep their admission result for internal use only. As a result, when applying to graduate programs, prospective applicants often lack the necessary information to gauge their competitiveness in the application pool. My report aims to address this problem by analyzing the Grad Cafe dataset and constructing several prediction models for chances of admission given the background of an applicant and their desired program. The prediction models build will be evaluated based on multiple metrics such as True Positive Rate (Sensitivity), True Negative Rate (Specificity), and Area Under the Receiver Operating Characteristic Curve (AUC).

1 DATASET AND SPECIFICATIONS

The dataset to be studied is from https://github.com/deedy/gradcafe_data. The only file of interest is all_clean.csv, it contains all admission record from Grad Cafe with clean university name. (e.g. NYU, New York University, and nyu are all merged under a single name)

all_clean.csv contains 271,807 entries and has the following key features:

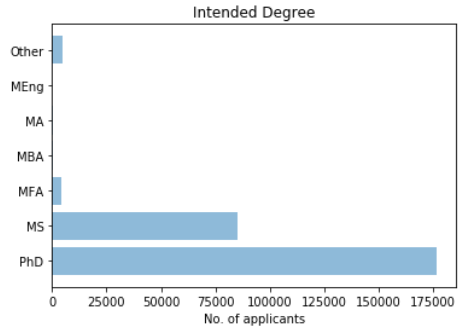
- uni_name: The name of the university.
- major: The intended major
- degree: The degree to be earned (MS, PhD)
- decision: The admission decision being reported
- ugrad_gpa: The candidate's undergraduate GPA
- gre_verbal: The candidate's GRE Verbal score
- gre_quant: The candidate's GRE Quantitative score
- gre_wrting: The candidate's GRE Writing score
- is_new_gre: Whether or not the candidate took the new GRE
- status: The residency status of the candidate (American, International)
- comments: comments of the candidate

1.1 Data Analysis

An exploratory analysis on the dataset was performed to reveal and visualize the basic characteristics of the dataset.

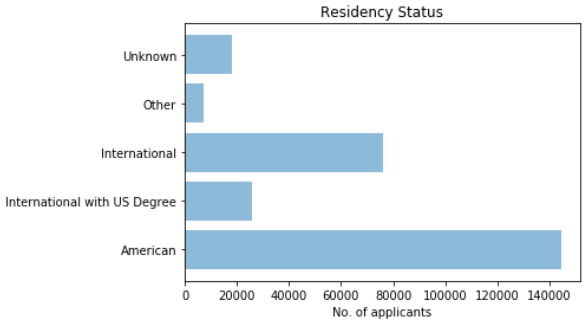
1.1.1 Applicants Profile

Among all the entries, 176,983 are for PhD admission, 85501 are for MS admission and the rest are for MBA, MFA, MA, and other non-degree programs. There are only 193 reported MBA applications, 114 MA applications, and 10 MEng applications.



Of all the applicants, 144,561 are Americans, 76,047 are internationals with foreign degrees, 25,867 are internationals with US degrees and the rest are not specified.

The mean GPA of all applicants is 3.660, with a standard deviation of 0.2827. The mean GRE Verbal percentile is 78.138, with a standard deviation of 21.387. The mean GRE Quantitative percentile is 70.788, with a standard of 22.725.



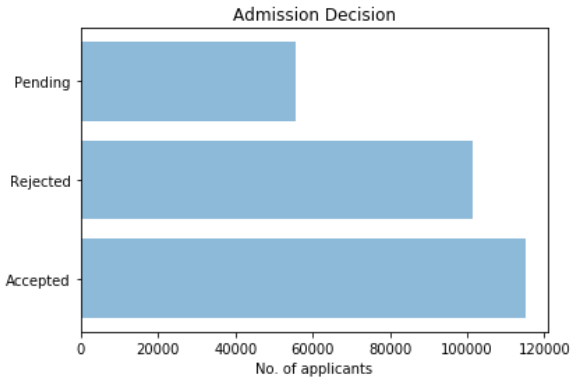
1.1.2 Most and Least Popular Majors

The task of finding the most and least popular majors is harder than expected. According to the author of the dataset, even though the university names are cleaned, the majors are not. Among the dataset there are 18,957 unique majors. After manually examining the data, I discovered that there are a lot of duplicates (Speech-Language Pathology vs Speech/Language Pathology), typos (mthmatics), abbreviations (MPH, OB&HRM) and names that do not indicate majors (Masters, graduate program). As a result, I first built a custom major stemmer. The algorithm is very simple: build a dictionary whose key is the cleaned major name and its corresponding value is a list of words that could be mapped to this major. Then transform the input major into lower case, search the dictionary to see if there is a match. For example, given {'Electrical and Computer Engineering': [‘ece’, ‘electrical', 'computer eng’, ‘robotic’]}, a reported major with name ‘ECE’ will be transformed to the cleaned name “Electrical and Computer Engineering”. This method is in no way exhaustive as it fails to consider many edge cases. But it successfully reduces the number of unique major to 2,918. For the least popular majors, only majors with > 10 applications are included. The result is as follow:

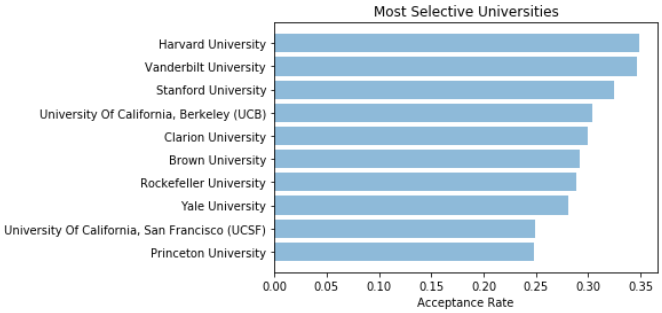
Most Popular Majors	No. Of Applications	Least Popular Majors	No. Of Applications
Speech Language Pathology	23088	Romance Studies	10
Computer Science	17810	Composition	10
Psychology	17721	Organizational Science	10
Political Science	16412	Forest Resources	10
Economics	16074	Integrated Studies	11

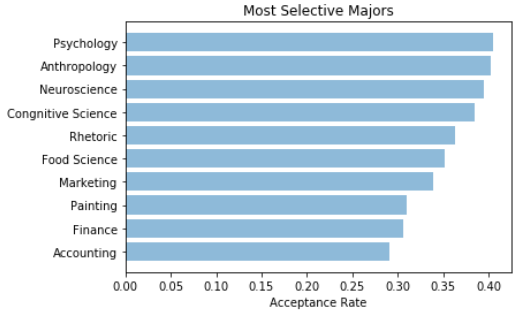
1.1.3 Selectivity and Test Scores

Among all reported admission decisions, 115,035 are acceptances and 101,311 are rejections, that gives us a global acceptance rate of 53.17%



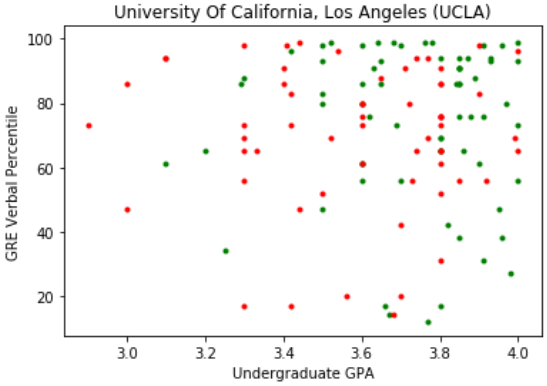
Of course, acceptance rates varies by intended major and university. According to the data, the most selective universities (with more than 100 applications) and majors are the following:

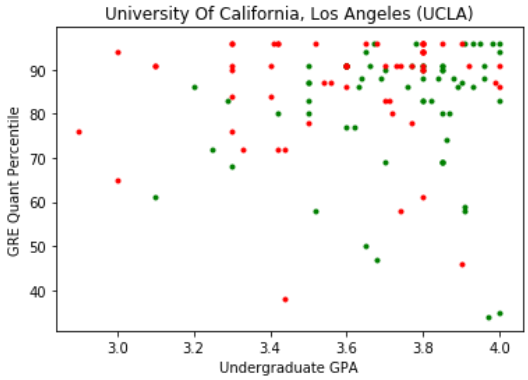




We notice that besides the usual Ivy school and elite public universities, UCSF and Rockefeller University both made it on to the list. Another finding is more concerning. According to the official statistics, Yale only has a 12% acceptance rate[1]; however, the dataset gives 25%, double that of the official rate. This discovery reveals the bias of this dataset: admitted students have higher chances to post their decision and their background online. Unfortunately, this limitation of the dataset cannot be easily addressed.

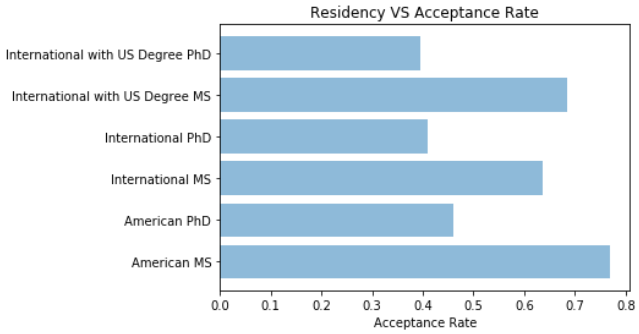
Now we gauge the effect of GPA and standardized test score on acceptance rate. Below is the distribution of accepted/rejected UCLA computer science MS and PhD applications (green: accepted, red: rejected)





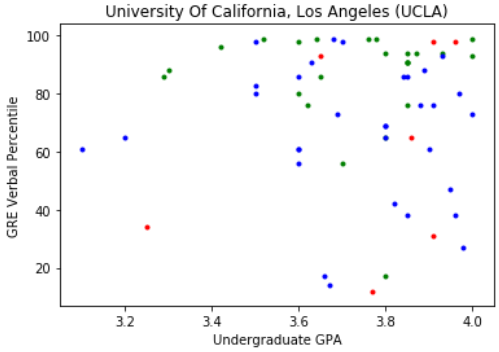
One can see that successful applicants typically have high undergraduate GPA (> 3.5) and high GRE Quantitative Percentile (> 80). Nonetheless, a high GRE Quantitative score does not guarantee your admission. A lot of applicants with top scores were rejected. On the other hand, the distribution of GRE Verbal is all over the place. The trend is the reversed on majors that are not math heavy.

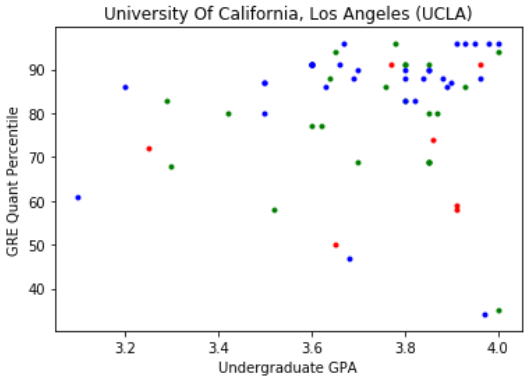
Residency also plays a role in influencing acceptance rate. As the figure below illustrates. American applicants have some advantage over international students, with or without US degrees. Notice that international students without US degrees seem to have a minor advantage over those who have US degrees, which is unexpected.



Another interesting finding related to residency status is the GRE score. Let's bring back the UCLA computer science admission example above, but this time, the distribution only contains accepted applicants. We can immediately see that even among those who were admitted, international students score considerably lower on GRE Verbal. That tells us graduate admission committees are more lenient toward international students in terms of GRE Verbal. This trend does not show up on the GPA vs GRE Quantitative graph.

(green: American, red: international with US degree, blue: international)





2 PREDICTIVE TASK

The goal of this report is to build a classifier that will predict whether an applicant can get into their desired graduate program given their scores and background. This is essentially a binary classification problem. We will first create some trivial predictor and gradually work up the accuracy of our model. We will mainly evaluate the performance of our models using Receiver Operating Characteristic (ROC), sensitivity and specificity. ROC measures true positive rate against false positive rate. The Area Under an ROC curve (AUC) measures the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [2].

The dataset is large but sparse, since most of the entries are self-reported by user on Grad Cafe. Out of the 27,1807 pieces of data, only 62,769 of them include GRE score and only 57,780 of them contain undergraduate GPA. Since it is impossible to make meaningful predictions with no features, the prediction models will only be trained and tested on the data where score, and other information are presented. There are 54,511 such entries. The dataset will be divided into training, validation and testing set to achieve optimal result.

2.1 Baseline

2.1.1 Guessing

One baseline predictor simply guess the label based on the distribution of testing labels. Intuitively, if the general population has X% chance of being accepted by their desired programs, then the chance of each individual of being accepted should be the same.

2.1.2 GPA Median

Another baseline predictor predicts true if the applicants' undergraduate GPA is higher than the median GPA of the admitted student of their target university and intended major, predicts false otherwise.

2.1.3 Validation Set Performance

Baselines	Sensitivity	Specificity	AUC
Guessing	0.5867	0.3962	0.4929
GPA Median	0.4201	0.6280	0.5245

2.2 Features

We first design and extract features then supply various combination of them to different models to test their accuracy. The features can mainly be divided in to three categories:

2.2.1 Numerical Data

The arguably most important features are the candidates' undergraduate GPA and GRE scores. Graduate schools receive huge volume of applications every year and they filter applicants first based on their GPA and standardized test scores. Unlike the holistic approach of undergraduate admission, graduate admission prefer candidates that demonstrate strong competence in their field of study. As revealed by the preliminary data analysis above, engineering majors prefer high GRE Quantitative score but do not care as much about GRE Verbal. Bearing this in mind, we have to normalize the candidate's GRE score according to the major they plan to apply. On the hand, we can normalize their GRE scores based on the university they are applying to. One thing worth noting is the dataset actually contains both old and new GRE scores. For obvious reason those two cannot be evaluated on the same scale, so I built two dictionaries that translate old and new GRE scores to percentile so they can be compared directly. The correlation between majors and scores disappear when it comes to GPA. After examining the dataset, I found there is no significant difference between GPA of admitted student between top universities and less selective universities. Of course this can be explained by the reputation of undergraduate institution the applicant attended. A 3.5 in elite college will definitely hold more weight than a 3.5 in no name college. Based on my personal experience, we can reasonably assume that GPA at foreign institution are valued less than American equivalent. However, there is no way to incorporate undergraduate reputation into the feature vector as the dataset does not have such information. Nonetheless, it is worth a try to normalize the GPA base on intended major and target university. Additionally, to compensate for the lack of undergraduate reputation, the acceptance rate of target university and intended major will be added.

The median score/GPA of admitted students can be obtained by constructing a dictionary of list, with universities and majors as keys, each time we encounter an acceptance, we append the GRE score, GPA to the list and we can easily find the median of the list using Numpy. The idea of calculating acceptance rate is similar, we construct a dictionary with universities and majors as keys. We append 0 to the list if we read a rejection and 1 if it is an acceptance. After that we use the sum of the list divided by the length of the list.

In summary, the numerical features will include acceptance rate of target university, acceptance rate of intended major, normalized GPA, normalized GRE Verbal percentile, normalized GRE Quant percentile and normalized GRE Writing score. All the fields are encoded as float into the feature vector.

2.2.2 Textual Data

Soft background is equally important in the admission process, as a stellar letter of recommendation and impressive conference paper can set two candidates apart. This is especially true for PhD admission where the candidate's ability to do research is heavily valued. A low GPA candidates with extensive research experience may well be preferred over one with high GPA and no relevant experience. To construct such feature, we have to look into the comments field of the dataset: around 55% of entires contains some text in the comments field so we have quite some texts to work with. However, I discovered that some people explicitly give out their admission decision or otherwise leaks information about their admission result (e.g. "I am excited", "tuition wavier"). If we do not remove those words from the dataset, we will defeat the purpose of admission prediction as the models can trivially make correct prediction based on information they are not supposed to know. To counter this problem, I constructed a list of forbidden word, any words in the list cannot be used as feature representation. I manually tweaked the list until the most frequent words in the comments no longer contain 'give away' words.

After data cleaning, we construct the feature using the bag-of-words model. To improve accuracy, all the stop words will be removed and all words will be processed by a stemmer. We first calculate the frequency of the all the words appeared in the entire dataset, then we identify the top 100 most frequent word occurred in entries where the applicants was admitted. Among the top 200 words are "fellowship", "research", and "internship", the finding confirms the assumption that extracurricular activities play a vital role in graduate admission. This feature will be encoded as a boolean list of length 200 and appended to the feature vector.

2.2.3 Categorical Data

According to the data exploration above, PhD programs on average have acceptance rate of 40%, compares to around 70% for MS programs. There is no reason to ignore this important variable. Another factor is residency. The data analysis suggests that American applicant can gain as much as a 10% rate boost comparing to international students with foreign degree. International students with US degrees also has minor advantage over international students without US degrees.

Both of those two features will be added to the feature vector by One-Hot Encoding. Residency and degrees can be conveniently accessed in the status and degree field of the dataset. For simplicity's sake, PhD will be encoded as [1,0]; MS and all other degree swill be encoded as [0,1]. Those degrees are very similar to MS in nature and the small sample size means they won't have much effect on the outcome anyway. Similarly, the residency feature will have 4 fields, representing American, international with US degree, international and other.

3 MODELS

There are 4 models of interest, all of them will have parameter class_weight set to 'balanced' due to the imbalance of classes in the dataset. I first tune and optimize each models to their optimal state then pick the one with the best performance.

3.1 Logistic Regression

Since we are performing a binary classification task. The first algorithm comes to mind is Logistic Regression, which maps a feature vector to a probability between 0 and 1 through the

sigmoid function. Logistics Regress provide informative outcome in term of probability; however, it is more susceptible to outliers than SVMs do.

3.2 Support Vector Machine

SVMs are also suited for classification problems. SVMs separate two group of data by finding a hyperplane that maximize their margin of separation. However, SVM is computationally intensive and can take hours to run on the dataset. As a result, I decided to use the LinearSVC from sklearn. LinearSVC is implemented in liblinear and has much better scalability. The trade off is LinearSVC does not provide the probability of one data point is positive.

3.3 Bagging SVM

Bagging is a technique that train multiple base classifiers on random distribution of the training data and aggregate them into a single classifier, which will predict a class label based on the popular vote of base classifiers [3]. Bagging reduce the variance of the base classifier thus improving accuracy. The bagging technique is implemented in sklearn as BaggingClassifier. I will bag the linear SVM above as the base classifier. I expect to see moderate increase in accuracy.

3.4 Random Forest

Random Forest Classifier is like a bagging classifier for decision trees, creating a forest of decision trees and aggregate the predictions of those subtrees. It scales easily and is unlikely to overfit. The sklearn implementation of Random Forest also come with a features_importance attribute that allow us to inspect the weight of different features [4]. One downside of the Random Forest is that it can be slow when predicting when doing prediction if there are two many decision trees as the Random Forest has to gather the result of all trees to make a prediction [4]. However, this is well worth it comparing to the training time of SVMs.

3.5 Optimization and Results

For Logistic Regression, the regularization parameter, and solver were tuned. The documentation says that saga solver is more suitable for large dataset; however, it gave me significantly worse result; no other solver can match liblinear. For the regularization parameter C, C = 1 yields the best result. I also tried different penalty term and L2 has a slightly higher accuracy than L1.

For the SVM, the main parameter needed tuning is regularization constant C. After experiments with different value. C = 0.1 produce the best outcome.

For the BaggingClassifier, I passed in the SVM with highest AUC as the base classifier and experimented with n_estimator: the number of base classifiers to be trained, max_samples: the maximum number of samples to be fed to base classifiers, and bootstrap: whether samples are drawn with replacement. It seems that there is almost no performance increase after n_estimator is above 200. After repetitive experiment, 200 estimator, drawing with replacement and a max sample size of 1/10 of the training set produce the best result.

For the Random Forest, there are two parameters that affect performance greatly. The first is n_estimator which control the number of base classifier and the second is min_samples_leaf, which control the minimum number of samples required to be at a leaf node (where the tree can make a decision). After experimenting with those two parameters, min_samples_leaf = 2 creates balanced sensitivity and specificity and the larger the n_estimator is the better the result is; however, the effect is minimal after n_estimator is passed 2000.

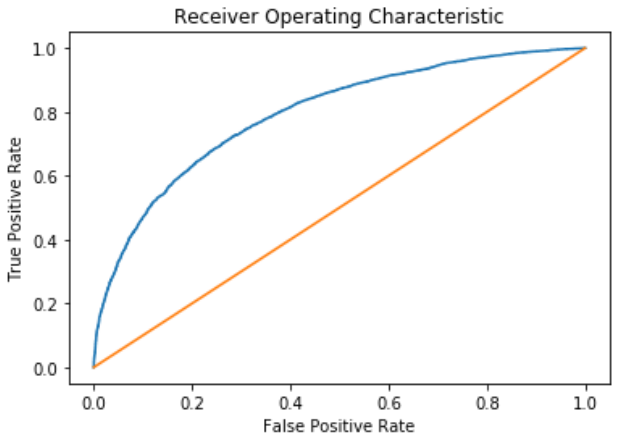
Below is a table comparing the performance of optimized models, all models except the Random Forest have low sensitivity and high specificity. If I drop the class_weight=balanced parameter the pattern is reversed. That means those models are either over optimistic or pessimistic when it comes to predicting admission chances, which is undesirable.

Models	Sensitivity	Specificity	AUC
Logistic Regression	0.6847	0.7512	0.7179
SVM	0.6633	0.7429	0.7031
Bagging SVM	0.6726	0.7503	0.7114
Random Forest	0.7201	0.7241	0.7221

The Random Forest Classifier with n_estimator=2000, min_samples_leaf=2 and class_weight=balanced is chosen as the final model due to its balanced sensitivity and specificity and its higher AUC. Below is the result on testing set:

Models	Sensitivity	Specificity	ACC	AUC
Random Forest	0.7156	0.7224	0.7270	0.7190

With a ROC curve:

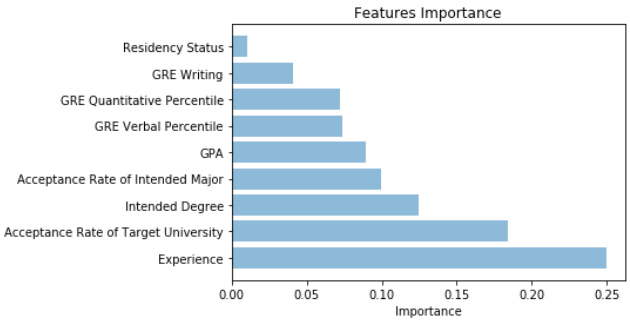


4 FEATURES IMPORTANCE

In this section, we explore the importance of features using the final model. Recall that there are three categories of features: (1) Numerical, (2) Textual, and (3) Categorical. We discard one of the feature and train the final model. The result is as follow

Features	Sensitivity	Specificity	AUC
(1)+(2)	0.6937	0.7118	0.7028
(1)+(3)	0.7758	0.5930	0.6844
(2)+(3)	0.6412	0.7545	0.6978

The importance of the soft and hard background (numerical and categorical) is very clear according to the table above. We can go further and break down the importance of each individual feature thanks to the features_importance attribute of the Random Forest Classifier:



It can be observed from the graph above that experience, acceptance rates and GPA are all good features representation. On the other hand residency status does not seem to affect the admission outcome much.

5 RELEVANT WORK

Predicting graduate school admission is not a new idea. Classification models have been deployed to assist graduate admission committees in numerous instances. For example to reduce the work load of graduate admission officers, The department of computer science at UT Austin developed a candidate evaluator, GRADE, which achieved an accuracy rate of 87%. More impressively, it identifies around 10% of admitted applicants with perfect precision and around 50% of rejected applicants with nearly no false rejections [5]. The GRADE model incorporates numerical, text and categorical data. For numerical data, GRADE encodes GPA and GRE a string (e.g. GRE Verbal > 80, GRE Verbal < 80), instead of raw percentile so the classifier can learn more complex trend. For text data, GRADE first preprocess the letter of recommendations of the applicants using stemmer and stop word removal, but applies Latent Semantic Analysis on the bag of words instead of calculating frequency. For categorical data, GRADE introduces several new fields such as the undergraduate reputation, applicant's research interest and preferred faculty advisor.

6 CONCLUSION

The analysis of the Grad Cafe dataset provide us some unique insights into graduate admission. However, as pointed out by previous analysis, the dataset has many limitations. The entries in the dataset are biased and with a lot of missing fields. Many useful features such as reputation of undergraduate institution and research interest are not presented, those constraints cannot be easily resolved unless more universities start releasing graduate admission statistics. With those limitations, my proposed model is able to achieve an accuracy of 73%. The series of experiments reveals that experience, acceptance rate, GPA and standardized test scores are among the most helpful features that can successfully predict the chances of admission for a particular candidate. Model-wise, Random Forest consistently out-perform logistic regressions and support vector machines in term of both overall accuracy and ROC. One possible reason is that there are some abstract trends of the data that logistic regression and SVMs are not able to pickup. If that is true, one should explore the possibility of training neural networks for graduate school admission to achieve better accuracy.

REFERENCES

[1] https://gradschool.princeton.edu/sites/gradschool/filesadmission_stats.pdf

[2] https://people.inf.elte.hu/kiss/11dwhdm/roc.pdf

[3] https://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

[4] https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

[5] Austin Waters, Risto Miikkulainen, "GRADE: Machine Learning Support for Graduate Admissions", http://www.cs.utexas.edu/users/ai-lab/downloadPublication.php?filename=http://www.cs.utexas.edu/users/nn/downloads/papers/waters.iaai13.pdf&pubid=127269