

Will you be admitted by your dream school?

刘益辰 2015111636

Abstract: Under the pressure of employment and the surge in the number of applicants for Graduate Entrance Examination(Kaoyan), as well as more and more families entering the middle class, an increasing number of undergraduates are eager to go abroad for postgraduate study, but choosing a school is a problem that must be faced by applicants. In this paper, machine learning models are constructed to help applicants predict whether they will be admitted to their dream schools by using the admission dataset of graduate students from the Management Information System of the United States, so as to better optimize the school selection list. In the model construction, this paper discusses a variety of basic models and constructs the ensemble model to enhance the result, and finally selects the model with the best performance in the validation set for super-parameter tuning, thus optimizing the model. In addition to modeling, this paper first cleaned the data and carried out Exploratory Data Analysis, random forest factor importance analysis and principal component analysis on the cleaned data.

Key words: machine leaning, data mining, model optimization, admission, MIS

1 Introduction

1.1 Background

Nowadays, more and more students are looking forwards to pursuing a higher degree abroad to expand their horizon and boost their background. Currently, Chinese students have account for a majority of the source of graduate programs in the US, and this trend is still going on. With the prosperity of Artificial Intelligence and Big Data, interdisciplinary programs which teaches both IT skills and other knowledge in other fields have becoming more and more popular. Students who desire to enter into a graduate program abroad are facing intense competition. Unlike the Graduate School Exam (Kaoyan) in China, Foreign graduate schools apply multiple standards to evaluate an applicant instead of judging only by the scores. Usually, Foreign graduate schools require applicates to submit the undergraduate GPA, English test scores (only for international students), GRE/GMAT scores, resume, personal statement, essays (which differ among schools) and recommendation letters from professors or leaders. Otherwise, applicants who had work experiences or research projects and have published papers will gain an edge in application, and some colleges might schedule an interview with applicants, which also affect the final decision. In order to optimize their application school portfolio in the application process, applicants should evaluate their probability of being admitted.

1.2 About the Dataset

The admission information is hard to acquire because graduate school rarely report specific information of their applicants. Thus, the only source of this kind may be the public forum. The dataset I use in this paper can be downloaded from Kaggle and the collector acquired the data from yocket. The dataset is about the admission of Master's in Management Information System (MIS) in the US's graduate schools. The dataset recorded people who either have applied to MIS programs or are just interested in those programs. From the perspective of my topic, I would just use the people who are admitted or rejected to make it a binary classification issue. When first opening the files, I find it quite messy because the information is either unstructured or incomplete. In order for me to build a model for this dataset, there must be a lot of preprocessing work to do.

1.3 The Goals of Research

- 1) I will try to build an optimal model to predict whether an applicant will be admitted by the graduate schools, a model which will at least beat the null model. The indicators I will use include accuracy, recall, precision and ROC(AUC). The anticipated accuracy is 75% on unseen values.
- 2) I will try to figure out the importance of attributes in the admission decision. According to the models, I will provide some suggestions for applicants to be more competitive.

1.4 Uniqueness

I minor in Information System and Management and I plan to apply to MIS graduate programs in the US, but none of prediction model exists to help applicants evaluate their probability of admission. Thus, I want to construct such a model to help both myself and other applicants. This topic is meaningful but rarely discussed in either periodicals or public forum. I conduct this research all by myself.

2 Data Analysis and Processing

2.1 Preprocessing

The number of university is quite a lot and some universities only have several observations, which might cause insufficient samples in building models, but I do not eliminate this attribute since which university you apply to is very likely to be an important factor of admission. Accordingly, I add the domestic ranking of US NEWS 2018 for each university and the MIS major ranking for its MIS program. The overall ranking is numeric while the major ranking is integrated into 3 tiers because many programs are not ranked.

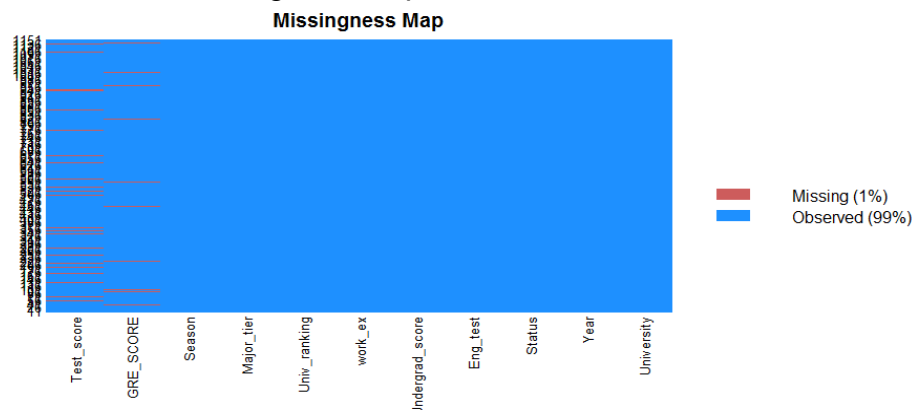
The dataset provides the application date which contains the year and the season, so I split it into two attributes---Year and Season.

English Test score is a bit tricky. Most took TOEFL but someone took IELTS and others took nothing. Since native speakers do not need to attend an English test scores to apply, I treat those not submitting English scores as native applicants. I assign 112 to the might-be native speakers' s English score.

What's more, I transfer the IELTS scores into TOEFL scores according to conversion table published in the internet. What's more, some people who have taken English exam but report 0 points in score, which are obviously outliers. I transfer these outliers into missing values.

Undergraduate GPA is provided in two kinds, one of which is followed by "CGPA" and another of which is in the percentage form. I use regular expression to standardize this attribute.

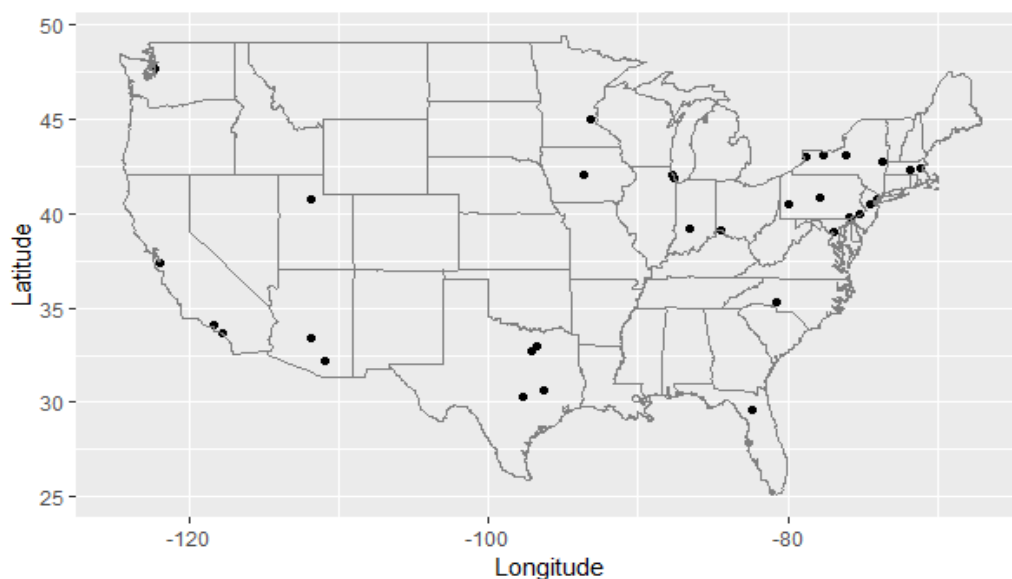
The dataset reports the months of work experience. It is certain that someone will apply for a master's degree without any work experience, so I treat the missing values as not having work experience.



At last, it is the time for imputing the missing values. I draw the plot to show the missing condition of the dataset, finding that only two attributes---"GRE_scores" and "Test_score" are missing. Fortunately, both have a missing rate lower than 10% and are numeric variables, so I can use KNN imputation to fill the missing values. The assumption is that similar applicants would have similar test scores. Before the imputation, I drop out the "Name", "University" and "Course", making all other attributes can be put into the imputation.

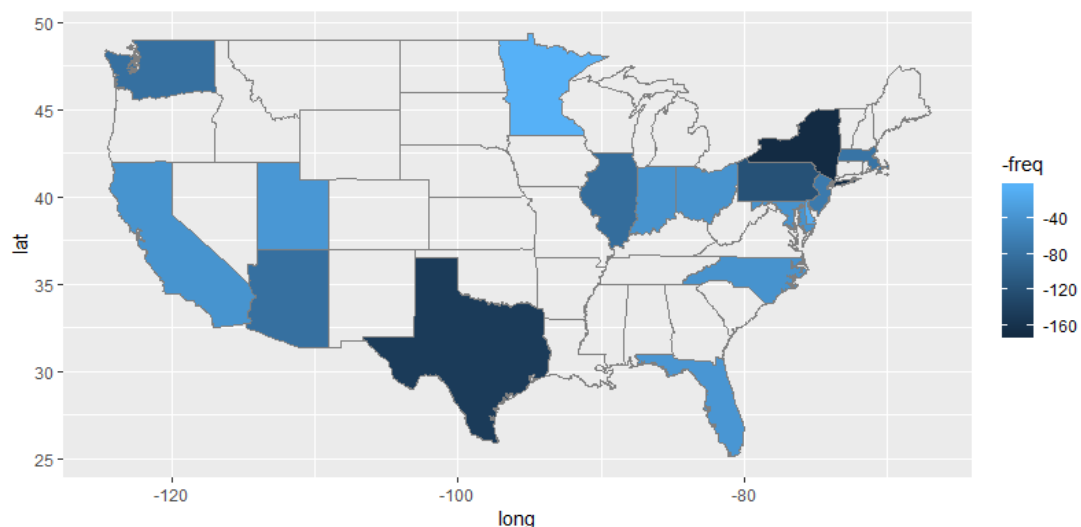
2.2 Exploratory Data Analysis

2.2.1 Geography



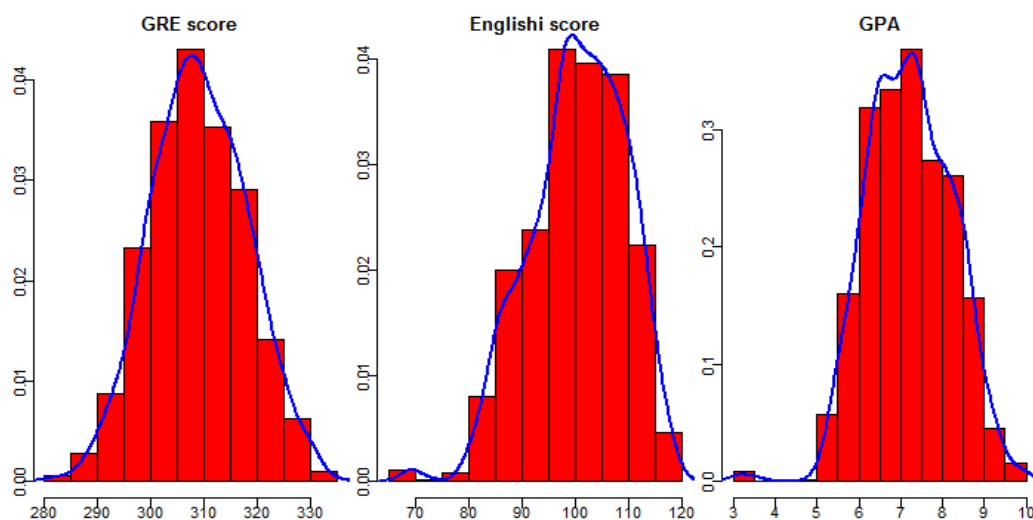
From the perspective of the geography, the universities that launch a MIS master's program concentrate on the northeast and southwest of United States. The distribution basically reflects the most developed area in United States. MIS is playing an increasingly crucial role in these regions.

The figure below shows the frequency of application in each state. Darker the color of a state is, more application the universities in this state have received. As shown in the figure, Texas and New York State owns the most application. Surprisingly, California where Silicon Valley is located does not receive as many applications as expected. Still, many states lack MIS application, meaning that talents are increasingly streaming into a minority of states where more employment opportunity can be acquired.



2.2.2 Scores Pattern

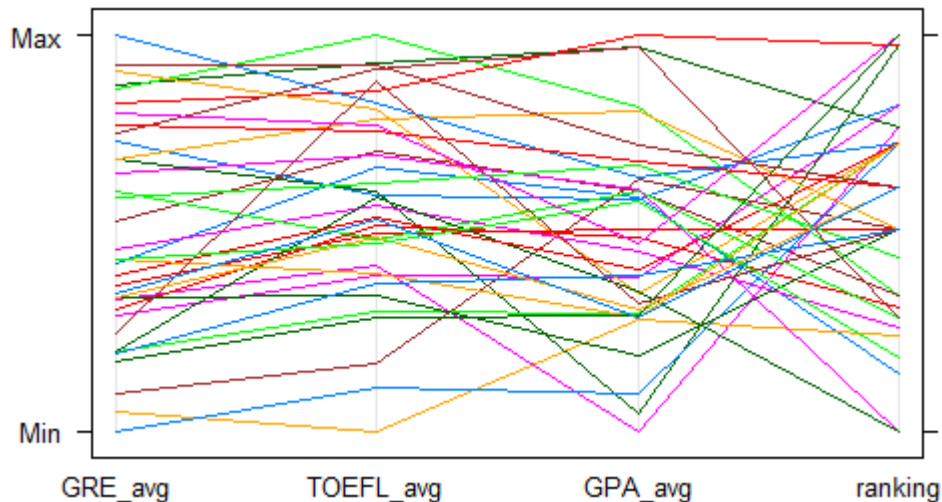
English test scores, GRE/GMAT scores and undergraduate GPA are often called “three dimensions” of an applicant. “Three dimensions” reflect the academic potential of an applicant and play a critical role, if not all, in the admission decision. First, I draw the histogram to show the distribution for all applicants. The “three dimensions” is basically distributed normally, with an average GRE of a about 307, an average TOEFL of about 100 and an average GPA of about 7.2.



Second, I explore the data of admitted applicants in terms of university. I generate the following table by using PivotTable in Excel. The table displays the average scores and standard deviation of each university. From this table, applicant can easily match themselves with the university. If your score exceeds the mean score, you might gain an edge over other competitors. If you have a score lower than average but higher than mean minus standard deviation, you are still competitive, but you are not prioritized to be admitted. If your score is lower than mean minus standard deviation, that would be like buying a lottery ticket.

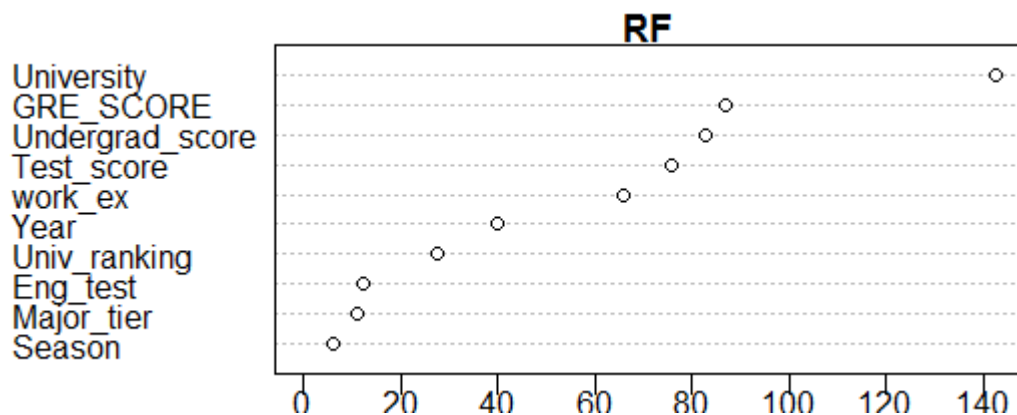
university	count	GRE_avg	GRE_sd	TOEFL_avg	TOEFL_sd	GPA_avg	GPA_sd	work_ex_avg	work_ex_sd	ranking	tier
Arizona State University	20	308.8	6.01	104.1	7.87	7.42	0.71	21.8	19.85	115	2
Boston University	8	306.8	7.72	99.6	8.5	6.43	0.88	22.4	25.37	37	3
Carnegie Mellon University	39	318.6	5.99	108.9	4	8.05	0.75	26.4	24.13	37	3
Drexel University	19	306.3	4.82	101.5	8.32	7.25	0.96	11.5	13.54	94	3
Illinois Institute of Technology	20	300.7	5.33	92.1	11.3	6.9	0.89	22.2	17.48	103	3
Indiana University Bloomington	20	318.4	5.78	110.1	5.88	7.8	0.95	11.3	12.6	90	2
Iowa State University	20	311.1	5.09	104.8	6.2	7.44	0.69	28.3	19.75	90	2
New York University	20	315.5	6.82	102.8	8.68	7.41	0.9	32.9	20.28	30	1
Northeastern University	20	305.9	7.89	99.2	10.12	7.08	1.06	20.5	24.23	30	1
Northwestern University	2	314.5	4.95	103	2.83	6.5	0.42	48	8.49	11	3
Pennsylvania State University	10	317.6	6.47	107.6	5.46	8.1	0.76	18	9.9	11	3
Rensselaer Polytechnic Institute	12	319.4	4.08	106.8	5.64	7.01	0.81	14.3	18.29	42	3
Rochester Institute of Technology	20	304	5.11	97.5	7.31	6.93	0.87	21.8	15.89	42	3
Rutgers University-New Brunswick	5	301.6	5.77	95.2	8.47	7.49	0.83	29	21.83	69	3
Rutgers University-Newark	20	299.6	5.95	94.1	6.57	6.59	1.06	23.3	20.44	42	3
Santa Clara University	20	309.6	7.13	102.4	6.3	7.19	1.31	24	22.45	100	3
Stevens Institute of Technology	20	306.9	7.74	98.3	7.16	6.75	1.04	14.2	18.32	69	3
Syracuse University	20	307.6	5.38	101.1	8.14	7.28	1.19	22.3	17.51	69	3
Texas A&M University-College Station	19	314.5	7.92	106.3	7.45	7.78	0.75	20.7	15.33	69	3
University at Buffalo SUNY	20	312.8	5.44	100.8	10.07	7.44	0.91	27.4	16.97	97	3
University of Arizona	20	316	7.23	108.6	4.62	8.05	0.98	23.6	18.18	97	3
University of California-Irvine	3	321.3	4.73	107	3.46	7.5	0.74	16.7	15.28	42	3
University of Cincinnati	20	313.7	5.17	104.7	7.34	7.45	0.96	30.7	18.75	133	3
University of Delaware	7	304	6.57	102.7	9.46	7.02	0.56	12.1	11.84	133	3
University of Florida	20	308.1	7.23	101.8	6.89	7.11	0.93	21.2	14.49	42	3
University of Illinois at Chicago	20	309	4.96	99.3	7.75	6.91	0.85	26.1	15.38	42	3
University of Maryland-College Park	20	312.4	4.18	103.4	4.98	7.55	0.73	19.5	17.89	78	3
University of Minnesota-Twin Cities	3	305	16.37	108	5.29	6.97	0.61	10	17.32	69	1
University of North Carolina at Charlotte	20	303.9	4.77	98.8	8.62	7.09	1.05	24.9	18.58	69	1
University of Pennsylvania	2	317	0	106	2.83	7.22	1.11	5	7.07	8	1
University of Texas at Arlington	20	303.4	5.05	97.3	9.49	6.92	0.88	21.2	19.7	8	1
University of Texas at Austin	20	316.4	5.89	105.8	5.27	7.57	0.95	30.4	27.25	56	1
University of Texas at Dallas	20	306.9	5.78	100.9	8.48	6.94	0.85	22.3	18.5	56	1
University of Utah	20	309	8.92	100.6	8.11	7.4	0.95	31	20.75	110	3
University of Washington	40	319.7	6.66	108.8	5.92	7.64	1.04	28.4	25.26	56	3
Worcester Polytechnic Institute	20	307.2	5.41	101.7	7.25	6.91	0.99	17.6	16.68	56	3

I use parallel plot to visualize the “three dimensions” with the university ranking. Generally, “three dimensions” stays in the similar levels for a university, indicating the university tend to choose those who can do well in all these three dimensions. However, the performance of “three dimensions” do not necessarily correlate to the university ranking. The higher in the ranking column, the higher the ranking of a university is. Some high-ranking university do not require high “three dimensions” of their applicants and vice versa. Since high “three dimensions” basically means intense competition, the ranking of the university appears not a significant factor applicants would take into account.



2.3 Feature Selection based on RF

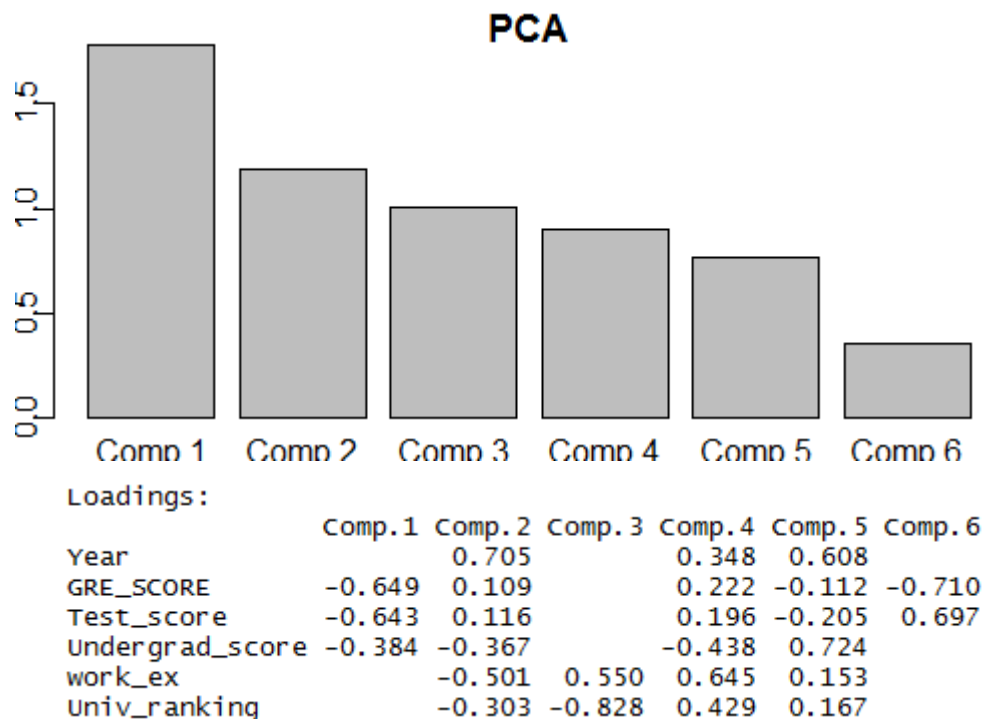
I use random forest to calculate the importance of each variables. The results are plotted in the following figure. Which university you applied make the most difference. “Three dimensions” also make huge effects. Among them, GRE is the most crucial, GPA is the second and the English test score is least important. Work Experience will also help with the application. The year you submit application and the university ranking contribute to the classification but not that much. The type of English test, major tier, season hardly have anything to do with the admission decision.



2.4 Principal Component Analysis

Principle Component Analysis is an unsupervised algorithm that cannot calculate non-numeric variables. Thus, I drop out the categorical variables and my classification label. The output is showed below. Since first 4 principal components have contribute over 80% of variance, we can just study the first 4 components. The fist principal component can be interpreted as a measure of “three dimensions”. The second principal component can be interpreted as a measure of work experience with application timing. The third principal component can be interpreted as the relationship between work and university ranking. The forth principal component is hard to explain. The results of PCA is generally consistent with the random forest importance, though relatively difficult to interpret. However, the importance among the principal components

differs not that much, so we will not reduce the any dimensions in the first place.



4 Models Building

4.1 Basic Model

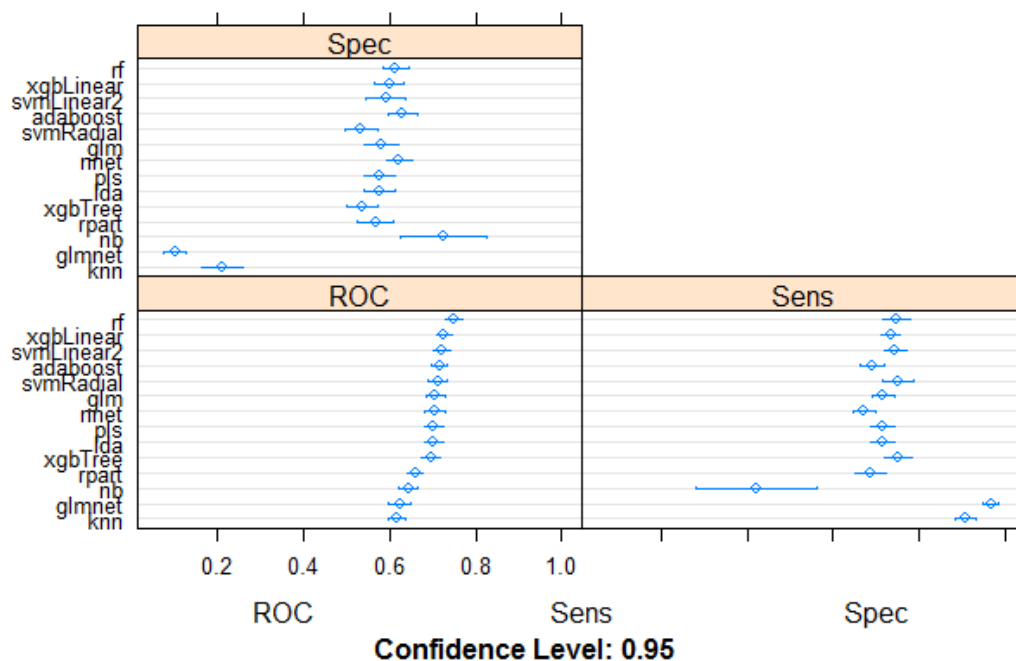
There are a lot of models that can be used in classification problems, including linear and nonlinear models. Linear models like logistic regression are generally more interpretable but less effective while nonlinear models are better but less interpretable, in which overfitting tends to occur. We set the benchmark on the basis of zero model and we are going to use a lot of different models here to see which one does better. Linear models include Logistic Regression(glm), Logistic regression with Regularization(glmnet), Partial Least Squares(pls), Linear Discriminant Analysis(lda) and Support Vector Machine with Linear Kernel(svmLinear2). Nonlinear models include Neuron Network(nnet), Support Vector Machine with Radial Kernel(svmRadial), K Nearest Neighbors(knn), Naïve Bayes(nb), CART Decision Tree(rpart), Random Forest(rf), Adaboost(Adaboost) and Xgboost(xgblinear/xgbtree).

I split the dataset into training set which accounts for 75% and validation set which accounts for 25%. I firstly train the models only by using the training set. In this way I can compare all models at one time. Although good performance in training set does guarantee good performance in validation set, bad performance generally means bad performance in validation set. Therefore, I can catch a glimpse of the basic models.

In model training, I scale the variables, and then use 4-fold cross validation and repeat the process 3 times. I randomize the super-parameters in each model to streamline the parameter tuning. Since this issue is a binary classification, ROC(AUC) is a suitable metric, so I use ROC(AUC) as main evaluation index.

The result of the basic models is lower than my baseline! The average ROC(AUC) range from 0.62 to 0.75. The random forest reap a ROC of nearly 0.75, higher than any other algorithms. The svmLinear and xgbLinear get ROC of about 0.72 and 0.73 respectively. Although the result has already been acceptable, I want to use ensemble to explore whether I can improve the model. Thus, in this case, the rf behaves best and the svm with linear kernel and gradient boosting follow. I want to enhance the result, so I will use ensemble learning.

ROC	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
nnet	0.5742904	0.6973411	0.7115612	0.7067061	0.7321118	0.7505174	0
glm	0.6357185	0.6910327	0.7028386	0.7075234	0.7408602	0.7518182	0
glmnet	0.5391041	0.5896654	0.6350532	0.6239366	0.6541416	0.6981061	0
pls	0.6326138	0.6878005	0.7018037	0.7031427	0.7310108	0.7503030	0
lda	0.6324660	0.6878005	0.7018037	0.7031231	0.7310108	0.7503030	0
svmLinear2	0.6507983	0.6998086	0.7374335	0.7216468	0.7498705	0.7622608	0
svmRadial	0.6447368	0.6778050	0.7232356	0.7123501	0.7386922	0.7944569	0
knn	0.5396215	0.5925460	0.6222727	0.6181964	0.6490828	0.6701515	0
nb	0.5770254	0.6148699	0.6648433	0.6452204	0.6817465	0.6927273	0
rpart	0.6134091	0.6342031	0.6582022	0.6604257	0.6806964	0.7132614	0
rf	0.6703134	0.7260428	0.7534743	0.7487597	0.7708860	0.8038143	0
adaboost	0.6531638	0.6964533	0.7242756	0.7157430	0.7410516	0.7656712	0
xgbLinear	0.6530160	0.7015421	0.7380383	0.7274304	0.7522508	0.7752809	0
xgbTree	0.6329095	0.6737879	0.6949761	0.6965768	0.7185216	0.7777942	0



Choosing the classifiers into ensemble is a crucial problem. The models in ensemble should not be highly correlated, otherwise we would end up predicting the same result as that when we use single models. Thus, my goal is to select models that not only perform well, but also have low correlation. From the table that provide the correlation among models, we can detect that four best classifier---rf, xgbLinear, svmLinear and adaboost, are not highly correlated. Therefore, I choose -rf, xgbLinear, svmLinear and adaboost into

ensemble learning.

	nnet	glm	glmnet	pls	lda	svmLinear2	svmRadial	knn	nb
nnet	1.0000000	0.5129336	0.3313589	0.5251864	0.5257333	0.540213467	0.3834417	0.43923520	0.3693724
glm	0.5129336	1.0000000	0.5748648	0.9945753	0.9945056	0.960867609	0.4132112	0.56970630	0.5679339
glmnet	0.3313589	0.5748648	1.0000000	0.5957044	0.5960042	0.626720101	0.3275027	0.70641487	0.6829325
pls	0.5251864	0.9945753	0.5957044	1.0000000	0.9999968	0.948167362	0.4075388	0.57679406	0.5502416
lda	0.5257333	0.9945056	0.5960042	0.9999968	1.0000000	0.948077268	0.4081364	0.57683453	0.5497021
svmLinear2	0.5402135	0.9608676	0.6267201	0.9481674	0.9480773	1.000000000	0.4260235	0.61760580	0.6422319
svmRadial	0.3834417	0.4132112	0.3275027	0.4075388	0.4081364	0.426023525	1.0000000	0.17917375	0.4887486
knn	0.4392352	0.5697063	0.7064149	0.5767941	0.5768345	0.617605796	0.1791737	1.00000000	0.7040467
nb	0.3693724	0.5679339	0.6829325	0.5502416	0.5497021	0.642231899	0.4887486	0.70404667	1.0000000
rpart	0.2142512	-0.1832678	-0.0805923	-0.2234572	-0.2230327	-0.009330662	0.3107372	-0.02852617	0.2402756
rf	0.6567408	0.5319877	0.4988300	0.5191775	0.5194224	0.617739782	0.7889650	0.35214873	0.6678497
adaboost	0.5992856	0.5679154	0.4816329	0.5581602	0.5586434	0.605023072	0.3397286	0.31878777	0.4734825
xgbLinear	0.4370174	0.2729228	0.3327162	0.2591196	0.2600290	0.346970715	0.6804629	0.10592954	0.5418583
xgbTree	0.4215721	0.4022275	0.4071217	0.3954111	0.3957180	0.511255291	0.6759247	0.23272931	0.5758041
	rpart	rf	adaboost	xgbLinear	xgbTree				
nnet	0.214251193	0.6567408	0.5992856	0.4370174	0.4215721				
glm	-0.183267829	0.5319877	0.5679154	0.2729228	0.4022275				
glmnet	-0.080592305	0.4988300	0.4816329	0.3327162	0.4071217				
pls	-0.223457187	0.5191775	0.5581602	0.2591196	0.3954111				
lda	-0.223032715	0.5194224	0.5586434	0.2600290	0.3957180				
svmLinear2	-0.009330662	0.6177398	0.6050231	0.3469707	0.5112553				
svmRadial	0.310737215	0.7889650	0.3397286	0.6804629	0.6759247				
knn	-0.028526166	0.3521487	0.3187878	0.1059295	0.2327293				
nb	0.240275582	0.6678497	0.4734825	0.5418583	0.5758041				
rpart	1.000000000	0.5077727	0.2631264	0.4230857	0.4518874				
rf	0.507772705	1.0000000	0.7587096	0.7576175	0.8346819				
adaboost	0.263126446	0.7587096	1.0000000	0.6045629	0.5607263				
xgbLinear	0.423085668	0.7576175	0.6045629	1.0000000	0.8066619				
xgbTree	0.451887392	0.8346819	0.5607263	0.8066619	1.0000000				

4.2 Ensemble

In this case, I use Stacking and greedy Bagging to ensemble my basic models. The following table shows the accuracy of ensemble models and basic models in training set and validation set for comparison with their ROC in the modeling process.

	rf	xgbLinear	svmLinear	adaboost	glm_ensemble	greedy_ensemble
Accuracy_train	99.76%	100.00%	69.55%	100.00%	98.05%	98.05%
Accuracy_test	72.89%	71.79%	68.50%	71.79%	74.73%	74.73%
Deviation Error	26.87%	28.21%	1.05%	28.21%	23.32%	23.32%
ROC(AUC)	0.7527	0.7317	0.725	0.7153	0.7581	0.757

From the table, we discover that glm stacking ensemble and greedy ensemble both enhance the accuracy in validation set by 2 percent though the accuracy in training set have been over 98%. It is worthy of mention that random forest, xgbLinear and adaboost nearly achieve 100 % accuracy in the training set, but not get a high accuracy in the validation set, with a deviation error over 26%. Thus, the random forest, xgbLinear and adaboost classifiers are overfitting. From the perspective of ROC(AUC), the AUC of glm ensemble increases to 0.7581, highest among all models. From the glm ensemble fitting coefficients, xgbLinear and adaboost are not significant. Ensemble models improve the overall results, but still not exceeding 75% of accuracy threshold. The results indicate that rf and xgbLinear have already overfitting. Although they seem to behave perfectly in the training set, they do not reap accordingly satisfactory performance in the unseen data. If we can acquire more data, the accuracy in trainset and accuracy in validation set are very likely to converge, thus making a better result. On the contrary, svmLinear classifier showcase low deviation error but low accuracy in training set either, meaning that it is still underfitting. If we can tune the parameter to let it better fit the training data, the accuracy in validation set is bound to increase significantly. For the next step, I need to eliminate the xgbLinear and adaboost and then tune the hyper-parameters of rf

and svmLinear to make better basic model.

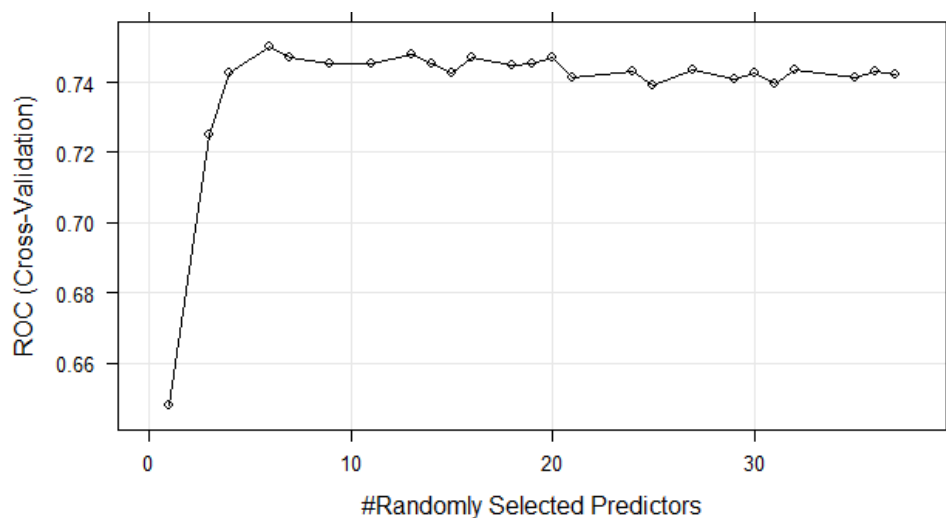
Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-3.1328	0.2842	-11.024	< 2e-16	***
rf	4.0781	0.5432	7.508	6.01e-14	***
xgbLinear	0.1523	0.2639	0.577	0.564	
svmLinear2	1.9670	0.3899	5.045	4.54e-07	***
adaboost	0.1537	0.8012	0.192	0.848	

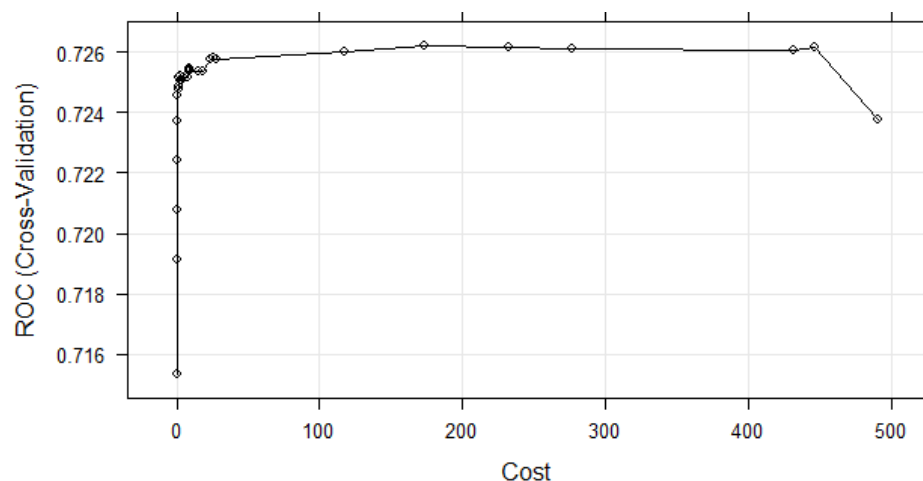
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

4.3 Hyper-parameter tuning

In this part, I customize the parameter for each model and still use 10-fold cross validation. In the random forest model, the “rf” method in “caret” package only need to tune the number of predictors in each tree. After iteration, the selected number is 5 and the correspondent ROC is 0.7500168, but different parameter does not generate huge variation of ROC. In this case, the accuracy on the validation set is 73.26%.



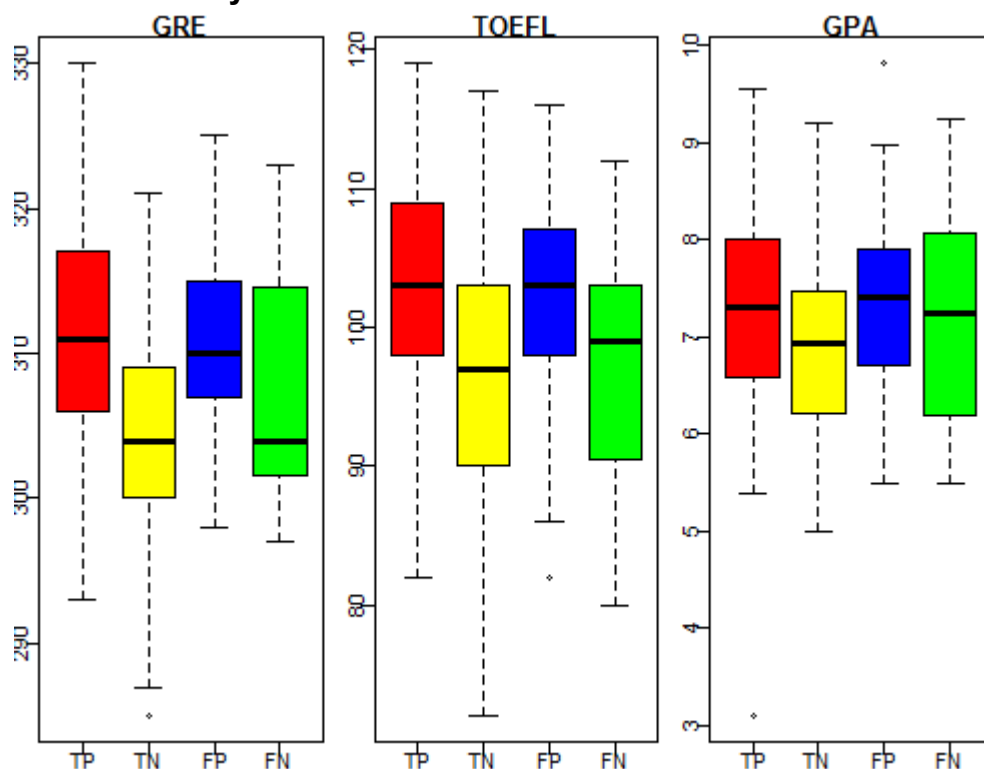
For the svmLinear, the only hyper-parameter is the cost of slack variable. The final value used for the model is cost = 173.8736, when the ROC on trainset approaches to 0.7262. The performance on the validation set is 68.5%, the same as the validation set. It seems that it is quite hard to boost the performance of a linear svm.



At last, I tune the parameter of the ensemble models containing the previous two classifiers. The summarized results are as followed. The ensemble models behave worse than they did before tuning hyper-parameters, but the ROC value does go up and exceed 0.76. The rf and svm have hardly improved performance either. The ROC of rf even decreased. As a result, we solely use neuron network as our ultimate prediction model.

	rf	svmLinear	glm_ensemble	greedy_ensemble
Accuracy_train	99.39%	69.18%	98.78%	98.78%
Accuracy_test	73.26%	68.50%	72.89%	72.89%
Deviation Error	26.13%	0.68%	25.89%	25.89%
ROC(AUC)	0.75	0.7262	0.7673	0.768

4.4 Mistakes Analysis



Since ensemble models achieve the highest ROC, I contrast the statistics of 4 parts in the confusion matrix generated by them. It reveals that from the perspective of “three dimensions” the TP are very similar to FP and TN are very similar to FN. FP applicants have a significantly high “three dimensions” than FN applicants does, indicating that high performance in “three dimensions” does not necessarily bring admission. The optimized model has already distinguished good from bad, but still failed to predict a fourth of applicants. I suppose that graduate school has a lot of more metrics for an applicant other than those in the models and these metrics are not highly correlated with the “three dimensions”; this possibility can account for the 1/4 mistakes rate.

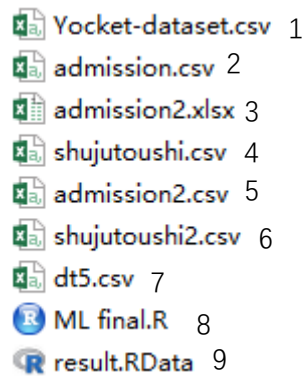
4.5 Conclusion

Based on the analysis above, I can conclude that due to insufficient number of observation and the inadequate ability of variables in the dataset to explain the result, we can only get the optimized model with accuracy of near 75%. However, this model is indeed able to significantly help applicants predict whether they can be admitted. The analysis result also tells us that graduate school does attach importance on the soft skills, which are not reflected in the model. One advice for the applicants is to make sure you do well in both academic experiences and soft skills.

5 Problems and Expectation

- 1) The number of observation is insufficient. Especially, each university only have about 40 application data. If fitted with complex models, it is very likely that the overfitting will happen. If we can acquire more data, the performance of validation will definitely go up to a high point.
- 2) The variables cannot fully explain the results. As we all know, graduate school admit an applicant not only through its scores but also based on his comprehensive performance including the essay, resume, personal statement and interview. Such factors are not only hard to quantified but also have a weak relationship with the standardized exam. If I can get these factors quantified and add them into the dataset, the results will be enhanced.
- 3) There may be some other approaches to pursue a better result. For example, better preprocess techniques can be applied. In terms of imputation, I use KNN to impute the missing values in this case. If using other methods like median, mean, MLE, multiple imputation or elimination, I may acquire a better result.

Appendix



Yocket-dataset.csv	1
admission.csv	2
admission2.xlsx	3
shujutoushi.csv	4
admission2.csv	5
shujutoushi2.csv	6
dt5.csv	7
ML final.R	8
result.RData	9

The figure above shows the files used in this report.

(1) is the original dataset.

(2) – (7) are the csv files that contains the transitional analysis contents

(8) is the all of codes of the research. The codes was completed in R studio and can be run in R 64x 3.5.1. The following packages are imported.

Rvest, tidyverse, ggplot2, lattice, caret, caretEnsemble, Rcpp, Amelia, DMwR, randomForest, stringr, maps.

(9) keeps track of the running results of the codes, including the established models.