The Comparison on Machine Learning Models An Example of Predicting the Probability of UCLA Graduate Admission

Sicong Li, Tianxin Zhang, Gefei Zhao

***Abstract*—This project focuses on predicting the probability of UCLA graduate admission. We view the topic as two supervised machine learning problems: regression and classification. The goal of the project is to compare the ability of different models in solving two kinds of problems. We use data of 500 UCLA graduate school applicants from Kaggle with seven features and one label (Chance of Admit) to analyze the problems. For the regression problem, we fit linear regression, Ridge regression, LASSO, elastic net, neural network models, then compare them over R^2 on test sets. For the classification problem, we use logistic regression, neural network, kernel SVM, random forest and decision tree to model the data and evaluate them by confusion metric, ROC curves and AUC values.**

***Index Terms*—Machine learning, regression, classification, Ridge, LASSO, elastic net, neural network, feedforward, SVM, random forest, decision tree.**

I. Introduction

THIS project is inspired by our group members's own experience on master program applications. There are many requirements and traits that the admission office looking for when a student wants to apply for master programs and each weighs differently in importance during the evaluation of admission decision to UCLA master programs. We are curious that how can we predict the probability of admittance to the UCLA master program of a candidate given all the information on GRE scores, TOEFL scores, university rating, statement of purpose, letter of recommendation strength, undergraduate GPA and research experience. The dataset on Kaggle has already sorted out the data of 500 applicants and we use the clean data to fit models and do further analysis. Our project is designed to apply different machine learning algorithms in two types of machine learning problems: regression and classification.

II. Task Description

A. Regression Problem

For the regression problem, there are five candidate models we can use to solve the problem. Linear regression and its regularized regression are capable of regressing the probability of admittance on seven provided features. The neural network model can give us the desired probability output as well. The task for the regression problem is to first fit the regression model. More specifically for regularized regression models, we need to find the optimal parameters through k-fold cross-validations; for feedforward neural network, we need to find the three optimal hyperparameters. Secondly, we use the model to make predictions on test samples. Finally, we choose the best model with the highest R^2 of five models over test sets.

B. Classification Problem

We converted the admission probability of each applicant into a binary classification based on the average admission rate of UCLA over the years. The binary label we generated was used to build the classification models. There are also five candidate models: logistic regression, neural network, kernel SVM, random forest and decision tree. For logistic regression, we need to use k-fold cross-validation when choosing the optimal parameter, set classification threshold manually, and generate the predicted label. Then evaluate the model via confusion metrics and ROC curves. For other models, we carried out different parameter comparison tests for each algorithm and selected the best parameter model through the comparison performance in the Validation Sets. Then we calculated the prediction accuracy of the optimal parameters in each algorithm, and generated Confusion Matrix.

III. Major Challenges and Solutions

To build both regression and classification models, we should use continuous labels and discrete labels for respective models. However, the original output labels which indicates the chance of being admitted are continuous variable between 0 and 1. So, the major challenge is to transform the continuous labels into binary labels of the admission decision of the school. Furthermore, the most challenging part is to determine a threshold of admission. This problem is not like any of the classification cases we did in homework or class lectures because the graduate school are not likely to let students enroll if their chance of being admitted is large than 0.5. But the admission office of graduate do not have a absolute standard of a ratio or number of students being admitted, it is hard for us to determine the exact threshold.

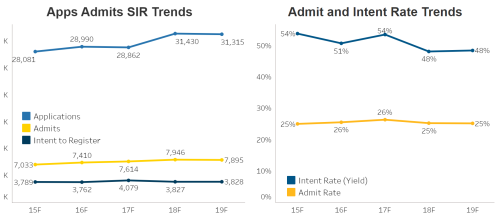

Fig. 1: Statistics of UCLA Admission

To solve the problem, we refer to the admission statistics of all programs of UCLA. Figure 1 shows the admission statistics of UCLA recent years.

The Admit rate is approximately around 25%. As a result, we could admit top 25% students by set the threshold to the 75% quantile of chance of admit. We wrote a decision function for probability of admission and we would firstly apply the function to our original dataset to get the binary labels of samples. Apart from this, we would apply the function to transform the prediction results of classification model into binary labels including logistic model and neural network, instead of using the default evaluation method of 0.5 threshold.

IV. Experiments

A. Dataset Description

The dataset is about admission system from a competition of Kaggle. The summary statistics of

TABLE I: Summary Statistics

	Number	GRE	TOEFL	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
mean	250.50	316.47	107.19	3.11	3.37	3.48	8.58	0.56	0.72
std	144.48	11.30	6.08	1.14	0.99	0.93	0.60	0.50	0.14
min	1.00	290.00	92.00	1.00	1.00	1.00	6.80	0.00	0.34
25%	125.75	308.00	103.00	2.00	2.50	3.00	8.13	0.00	0.63
50%	250.50	317.00	107.00	3.00	3.50	3.50	8.56	1.00	0.72
75%	375.25	325.00	112.00	4.00	4.00	4.00	9.04	1.00	0.82
max	500.00	340.00	120.00	5.00	5.00	5.00	9.92	1.00	0.97

the features and label are shown in table I.

1) Features vs. Continuous Label: From the aspect of regression, the relationship between input and output are significant in general which is shown in figure 2. Specifically, we can see that the most influential features of applicants are GPA, GRE and TOFEL which can strongly reflect the comprehensive strengths of students. Then graduate schools would focus on the application materials including SoP, letters of recommendation and university rating. The minor feature of students considered by admission office is the research experience. It might because undergraduate students is hard to work on scientific research.

2) Features vs. Binary Label: From the aspect of classification, we can visualize the features with binary label. We can look at the strip plot and scatterplot between features and label.

The figure 3 and figure 4 are strip plot for categorical features including university rating, rating of statement of purpose, the rating of letter of recommendation and whether students have research experiences. In general, better performance in these evaluation indices would lead higher portion of admission. Specifically, there are few admission

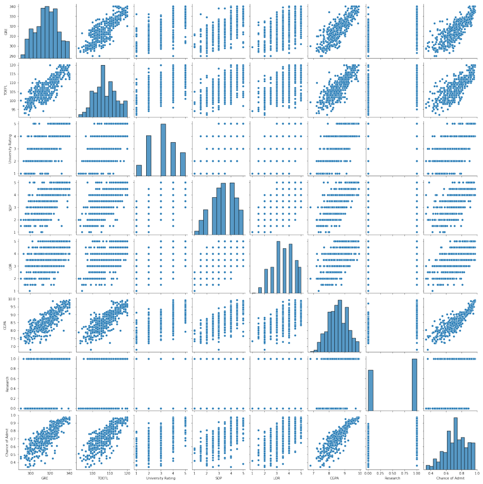

Fig. 2: Pair Plot

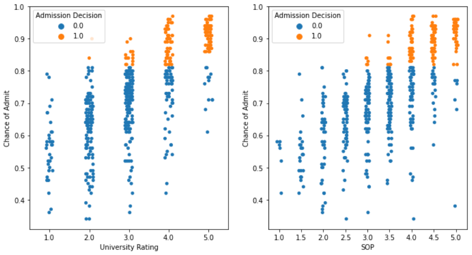

Fig. 3: Strip Plot

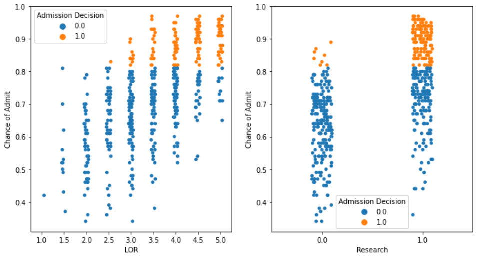

Fig. 4: Strip Plot

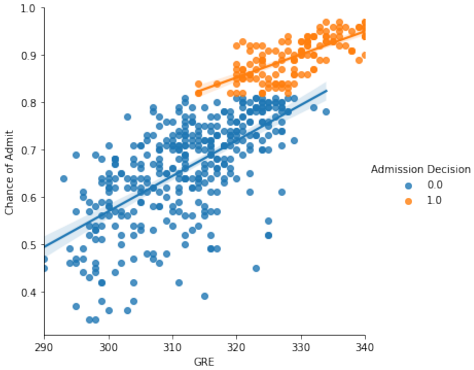

Fig. 5: Scatter Plot of GRE

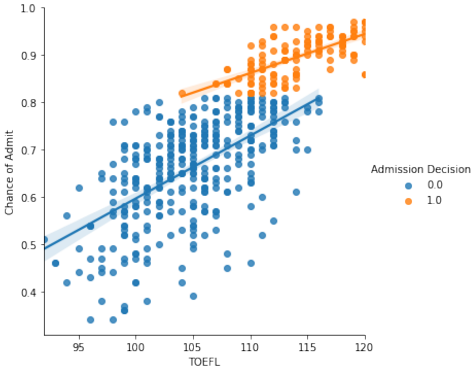

Fig. 6: Scatter Plot of TOFEL

for university rating under level 3, rating of SoP less than 3 and rating of recommendation letters less than 3. Besides, approximately more than 85% student admitted has research experience.

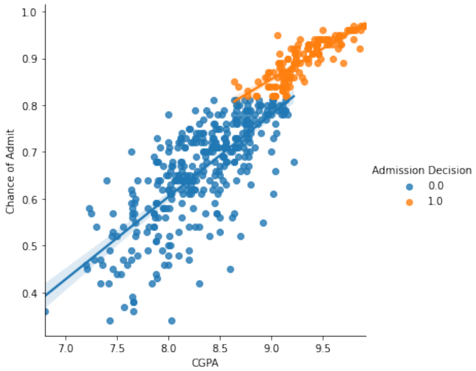

Fig. 7: Scatter Plot of GPA

For continuous variables including GRE, TOFEL and GPA, the scatter plots of features vesus chance of admit by admission decision show significant distinction boundary for rejecting application or admitting overall. Students with GRE scores greater than 320, TOFEL scores greater than 110 and GPA greater than 8.5(full credit is 10) have really large probability to be admitted.

B. Evaluation Metrics

1) Regression Model: We use the Scikit-learn package's build-in linear regression, Ridge regression, LASSO and elastic net functions to set up models and predict the probability of admittance to the UCLA master program. We build up models on training sets and use 5-fold cross-validation R^2 evaluation on validation sets to choose the optimal value of parameters used in the regularized regression model.

a) Linear Regression: The first model is the simplest linear regression model, after rescaling on features, we regress the Chance of Admit on seven features using training samples, which gives the following linear regression

TABLE III: LASSO Cross Validation Result

	α = 0.01	α = 0.001	α = 0.0001
validation R^2	0.4974	0.4965	0.4964

model: $\widehat{\text{Chance of}}$ Admit = 0.6740 + (0.0619)GRE + (0.0443)TOEFL + (0.0100)University Rating + (0.0041)SOP + (0.0418)LOR + (0.1648)CGPA + (0.0112)Research. The corresponding R^2 on validation set is 0.66.

b) Ridge Regression: The second model is the Ridge regression model, which is a linear regression model with L2 regularization. The reason why we include regularized linear regression models is that our sample size is relatively small, and the size of training sets is even smaller after data splitting. To avoid the overfitting problem, we choose several regularized regression models. The parameter α needed to be select through cross-validation. In table II, we can see that the optimal α = 1, which gives the highest mean validation R^2.

c) LASSO: The second regularized regression is a linear regression with L1 regularization, which adds an L1 norm to the original cost function to create a penalty term on overfitting. In table III, we summarize the result of 5-fold cross-validation when we choose parameter α in LASSO. α=0.01 gives the highest mean validation R^2 of 0.4974 and is chosen to be the best α used in the LASSO model.

d) Elastic Net: The last regularized regression model is Elastic Net with L1 and L2 norm added in the cost function. The parameter needed to choose is k from k-fold cross-validation. From a range of k values [3, 5, 7, 9, 10, 50, 100], we fit elastic net models on the training set find the optimal k value is 3 and the respective validation set score is 0.6645.

TABLE IV: Validation result for Regression FNN

R^2	in validation set	in test set
FNN1	0.6417	0.8106
FNN2	0.6752	0.7826
FNN3	0.6243	0.8106

e) FeedForward Neural Network: We tried three different groups of parameter Settings. The first model had one hidden layer with 50 neurons and used the ELU as the activation function. For the second model, we set 2 hidden layers, which has 10 neurons in the first layer and 80 neurons in the second layer. Both of these two layers used ReLU as activation function. For the third model, we simplify the structure of the second model. The third model still has 2 hidden layers, but the number of neurons in the first layer is reduced to 5 and ReLU is still used as activation function, while the number of neurons in the second is reduced to 10 and replaced with ELU as activation function. 0.01 is used as the learning rate and MSE is used as the loss function in all the three groups of models. The third model had the best performance in the Validation Set, so we used the parameters of the third model to predict in the Test Set. See table IV.

2) Classification Model:

a) Logistic Regression: With the help of 5-fold cross-validation, we find out the optimal C (Inverse of regularization strength) in the built-in logistic regression function that generates the highest mean validation accuracy are 1 and 10. We choose the smaller C, 1, to make the later prediction on testing sets simpler. The cross validation result

TABLE II: Ridge Cross Validation Result

	alpha = 10	alpha = 1	alpha = 0.1	alpha = 0.01	alpha = 0.001
mean validation r2	0.53	0.54	0.51	0.50	0.50

TABLE V: Logistic Regression Cross Validation Result

	C = 0.01	C = 0.1	C = 1	C = 10	C = 100
mean validation accuracy	0.74	0.82	0.96	0.96	0.94

TABLE VI: Validation result for Classification FNN

Accuracy	in validation set	in test set
FNN1	1.00	0.912
FNN2	0.96	0.87
FNN3	0.96	0.88

of logistic regression is in table V

b) FeedForward Neural Network: We built three models with the same structure as Regression FNN, but here we used cross-entropy as the unified loss function. Since our threshold is set to 0.75, not 0.5 default setting in the Keras package. We converted the output into the probability from 0 to 1, and then classify it through our own threshold setting. Finally get our binary prediction result and compared them with Ground Truth Label. We used these three models in Validation Set and all got quite good results in table VI. The first model even reached an accuracy of 1 in Validation Set. So we used the first model parameters in the Test Set.

c) Kernel Support Vector Machine: To build a SVM model, we firstly use cross validation to determine the optimal hyperparameter. The figure 8 show the result of cross validation and the optimal hyperparameter set is C=1 and kernel=linear.

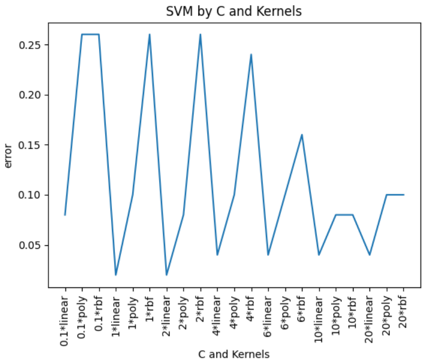

Fig. 8: Cross Validation Results of SVM

d) Decision Tree: To further develop models for graduate admission system, we try to use decision tree algorithm. The main idea of decision tree is to incrementally updated by splitting the dataset into smaller datasets (both numerical and categorical) to make prediction of label. We just use the decision tree function with default parameters.

e) Random Forest: To dive deeper into this problem, we apply random forest algorithm on our data. Random forest is kind of bagging of decision tree model. But it is different with both of them because not all the features in dataset are used to train model, thus some over-fitting features can be avoided and no pruning of the decision tree will be performed. We use the random forest function

TABLE VII: Summary of Regression Models

Models	R^2
Linear Regression	0.8278
Ridge Regression	0.8278
LASSO	0.8027
Elastic Net	0.8277
FeedForward Neural Network	0.8106

TABLE VIII: Summary of Classification Models

Models	Accuracy
Logistic Regression	0.912
FeedForward Neural Network	0.912
Kernel SVM	0.912
Decision Tree	0.908
Random Forest	0.920

with default parameters.

C. *Major Results*

1) Regression Model: The summary of evaluation results of regression models is concluded in table VII.

2) Classification Model: The summary of evaluation results of classification models is concluded in table VIII.

The confusion matrix of logistic regression, neural network and SVM is shown in table IX, X and XI respectively.

TABLE IX: Confusion Matrix of Logistic Regression

	$\hat{y} = 1$	$\hat{y} = 0$	Recall
y=1	176	11	0.94
y=0	11	52	0.83
Precision	0.94	0.83	0.912 (Accuracy)

TABLE X: Confusion Matrix of FNN

	$\hat{y} = 1$	$\hat{y} = 0$	Recall
y=1	176	11	0.94
y=0	11	52	0.83
Precision	0.94	0.83	0.912 (Accuracy)

TABLE XI: Confusion Matrix of SVM

	$\hat{y} = 1$	$\hat{y} = 0$	Recall
y=1	182	5	0.97
y=0	17	46	0.73
Precision	0.91	0.90	0.912 (Accuracy)

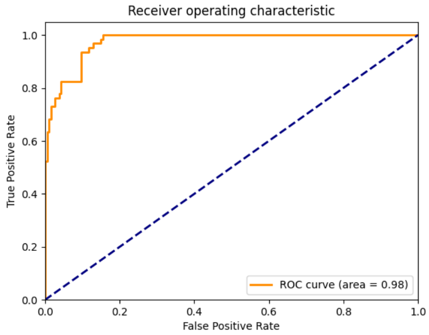

Fig. 9: ROC of Logistic Regression

D. *Analysis*

1) Regression Model: The finalized model with optimal parameters is applied on testing sets, then we conduct the R^2 to compare performance between different regression models shown in table VII. We can see that five models perform similarly in this regression problem. The R^2 on testing samples are all around 0.80, in which linear regression and Ridge regression have the highest among five models, while LASSO has the lowest.

Besides, to test whether our FNN model is

TABLE XII: Results of Bootstrap of SVM

Mean	0.9318
Standard Deviation	0.0209

robust we performed bootstrap to the optimal regression FNN model. The R^2 have dropped significantly from the original 0.81 to a negative number, which mean the FNN model may have overfitting problem, so different sampling results lead to huge differences in results.

2) Classification Model: We could see Logistic Regression, FNN and SVM generated very similar confusion matrix in table IX, X and XI. Meanwhile, Decision Tree and Random forest also perform well. The accuracy of all models reached above 0.9, in which the Random Forest algorithm has the best performance of 0.92 accuracy.

Without a fixed threshold, the ROC curve from logistic regression model shown in figure 9 is close to perfect curve with an AUC value of 0.98. We conclude that the logistic regression model performs well under all possible threshold.

In order to avoid the model instability caused by the samll sample size, we did bootstrap in the FNN and SVM models. The result of bootstrap of SVM model is shown in table XII, It shows SVM model is robust and it has good generalization ability. But in FNN model, the accuracy dropped sharply to 0.57. We discussed this problem further in Further Work Section.

V. Conclusion and Future Works

We experimented with different machine learning algorithms for these two types of problems (regression and classification). Almost all of the algorithms performed quite well. Basically, for regression problem, all of the adjusted models can reach R^2 above 0.8, among which linear regression and ridge regression have the highest R^2 with 0.8278. For classification problems, all the models have very high accuracy, which could achieve an accuracy rate above 0.9. And the random forest model achieved 0.92, which is a very good prediction effect. Overall, the model based on machine learning can predict the admission probability of UCLA applicants accurately.

Reviewing the whole project, the sample size of our dataset was very limited due to the data obtained from Kaggle. Therefore, our data and prediction models may not be fully effective in representing all applicants. Based on this problem, we did bootstrap in FNN and SVM models respectively. In the SVM model, there is still a good accuracy after bootstrap, but there is a big difference between the results of the FNN model after bootstrap and the initial test set. Although we sampled these 500 data at random, that could only make the distribution of our sampling more similar to the population, but the sample set itself was still the original 500 samples, so the sample size did not increase significantly. The future work may focus on how to expand the data sample size, so that our model can more accurately reflect UCLA's admission preferences.

Acknowledgment

The authors would like to thank Profess Liu and teaching assistants Manvendu Navjeevan and Onyancha Onyambu for their teaching and supportive help.

References

[1] Dataset: 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

[2] UCLA graduate program statistics: https://grad.ucla.edu/graduate-program-statistics/admissions/?t=Annualsnapshot

[3] Third party packages: numpy, pandas, seaborn, sklearn, matplotlib, tensorflow, pylab