ML4774 Graduate School Recommendation Project .

Technical Report

Tianze Ren
tr2bx@virginia.edu

Kaicheng Chu
cbf6ah@virginia.edu

Haley Fitch
hnf8bgt@virginia.edu

December 10th, 2023

1 Abstract

The primary aim of this project is the creation of a recommendation system designed to provide prediction to graduate program admission decisions, thereby narrowing the information gap and aiding administrative decision-making. The system will leverage a multi-criteria machine learning algorithm to evaluate and match applicants' profiles with program prerequisites. The expected outcome is an enhancement of the application process for prospective students, providing a streamlined and targeted approach to navigating the complexities of graduate program admissions. The anticipated benefits of such a system include time and resource efficiency for the applicants and an improved mechanism for programs to identify and attract candidates who fulfill their specific criteria.

2 Motivation

When it comes to their future moves after college, university students tend to have different plans. Some might decide to start their own business, some might want to work for big companies, while others choose to attend graduate programs to keep improving themselves. Information asymmetry would be a severe obstacle for graduate program applicants since the compatibility between students' status and school requirements is hard to measure. Every applicant has different educational backgrounds and professional experiences, and every program looks for different qualities in applicants. Thus, admission decisions are hard to make. We would like to build a graduate program recommendation system, whcih will provide applicants with an insightful reference for their application decisions.

3 Hypothesis

With a background of graduate program admissions, our team expect that applicants with higher academic scores and more extensive research or industry experience are significantly more likely to be admitted. This hypothesis is grounded in the assumption that these three factors are the ones the schools pay most attention to, and we believe these three are critical indicators of a candidate's potential for success in advanced study. The project aims to empirically validate this assumption by employing a machine learning classification system that analyzes various aspects of applicants' profiles. The broader significance of this study lies in its potential to enhance the understanding of admission criteria efficacy and fairness, thereby contributing to the development of more effective and equitable graduate admission processes for the graduate admission office as well as providing helpful information for students who are interested in the graduate programs.

4 Literature Review

1. 'Asymmetric Information, Parental Choice, Vouchers, Charter Schools and Stiglitz'
The paper discusses the impact of parental choice on the quality of education and the public good. It highlights the importance of considering factors such as race, religion, wealth, and ethnicity in school choices, as these can lead to "imperfect competition" and a decline in educational quality. This paper gives insight into the complexity of the

education application process, indicating lots of social and economic factors at play.

2. 'Food for Thought: Information Asymmetries and the Market for Higher Education'
This article reveals the current situation of information asymmetry in the market of higher education. Schools know every detail of the students and their family conditions, while students are more uninformed about the admission requirements. This is exactly the problem we are trying to solve.

5 Methods/Data Resources

Machine Learning Models applied: K-nearest-neighbors; Logistic Regression; Naive Bayes; Support Vector Machine with rbf, sigmoid, and polynomial kernels; GridSearch and Decision Tree; Random Forest; Neural Network; LightGBM; Voting System;

The chosen machine learning algorithms for the recommendation system are aimed at capturing the nuanced relationship between graduate program requirements and applicant profiles. K-Nearest-Neighbors (KNN) is selected for its intuitive approach to find similar applicants within the admission landscape. Support Vector Machine (SVM) with different kernels—rbf, sigmoid, and polynomial—is employed to dissect and interpret complex, nonlinear patterns in high-dimensional data, which is typical in educational background and experience datasets. GridSearch is utilized in conjunction with Decision Trees to methodically search for and optimize the best parameters for our models, ensuring the most accurate predictions in matching applicants to their ideal graduate programs. Each algorithm brings a unique strength to the system, from simplicity and interpretability to powerful pattern recognition and optimization capabilities. More complecated models such as random forest, voting system, and neural network are used to strive for higher accuracy models and explore the deeper connection between students and their admitted programs.

Data url: link to both datasets used (changed "original data" to "graduation recommendation program")

Data description: this dataset includes information about graduate program applicants and their applied universities admission results (admitted/ rejected). The dataset contains two major files:

1. Original.csv: profiles of students with admits/rejects to 45 different uni- versities in USA.

2. score.csv contains score conversion for GRE old to GRE new score mech- anism.

Potential Used Columns: major, researchExp, specialization, tofleScore, de- partment, greV, greQ, Cgpa, univName, admit

Dataset size: 11.43 MB
Total attribute: 26
Total observation: 14798

6 Data Processing

Data Processing
The dataset originally contains 26 attributes and 53645 observations. Attributes only indicating personal information are dropped, attributes with overlapping information are dropped, and attributes with too many missing obervations are dropped.
Below are the detailed processing decisions.
1. Feature 'userName' is dropped
2. Feature 'toeflEssay' is dropped.
3. Feature 'specialization' is dropped for containing complicated categorical values and too many na values
4. Feature 'department' is dropped for containing too many overlapping variables and containing overlapping information with 'major'
5. Feature 'userProfileLink' is dropped
6. Feature 'greA' is dropped
7. Feature 'topperCgpa' is dropped
8. Feature 'termAndYear' is dropped
9. All feature for gmat are dropped
10. Feature 'journalPubs' with missing values are filled with 0

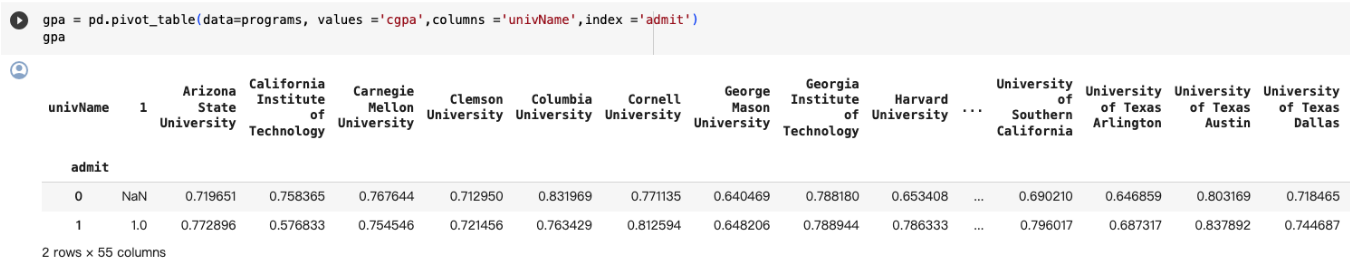
11. Missing values in 'ugCollege' are dropped
12. Observations with empty feature 'major' are dropped
13. Two features 'greV' and 'greQ' are converted using the same standard
14. Feature 'cgpa' is converted to the same standard using 'cgpaScale'

The whole dataset is split into a test set, validation set, and train set. Then, a data processing pipeline is built to normalize numeric variables using Standard Scaler and to split categorical variables into binary variables using One Hot Encoder.
One limitation of using One Hot Encoder in our case is that it generates too many binary variables (over 1000 variables in total), which makes the model training process extremely slow, especially for Decision Trees, and produced the problem of multicollinearity. Thus, we propose the use of word embedding to process the categorical variable 'major', which contains confounding values with overlapping information such as '(MIS / MSIM / MSIS / MSIT)', 'Computer Science', and 'Biotechnology/bio-engineering'. We grouped the semantic meaning of all values into 30 groups, preversing the diversity of different majors while reducing the dimensionality after applying the One Hot Encoder. We stick to One Hot Encoder for other categorical variables with the following reasons:

1. Variable 'program' has only 5 distinct values, which is suitable for One Hot Encoder
2. Variable 'ugCollege' and 'univName' has school names without semantic relationships between each other. Thus we still use One Hot Encoder for these two variables.

7 Data Analysis and Interpretation

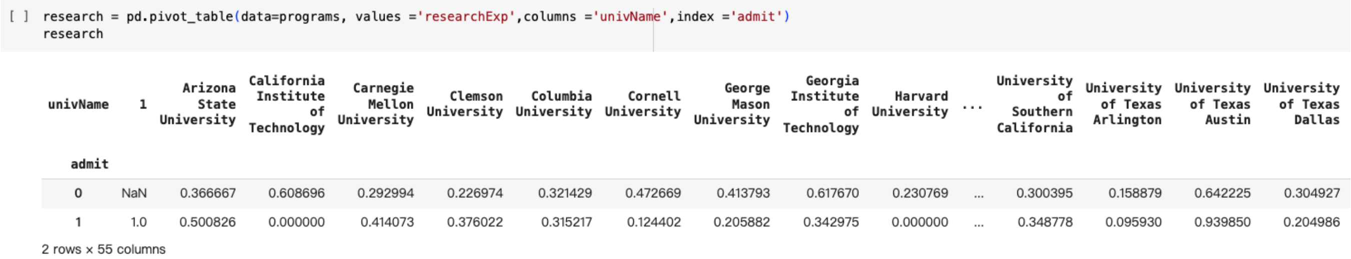
Conditional on the admission decision, we found out that the average values of those key attributes such as 'cgpa', 'researchExp', and 'industryExp' do not have great distinctions, which indicates that there are other important features at play outside of this data set.



```
gpa = pd.pivot_table(data=programs, values ='cgpa',columns ='univName',index ='admit')
gpa
```

univName	1	Arizona State University	California Institute of Technology	Carnegie Mellon University	Clemson University	Columbia University	Cornell University	George Mason University	Georgia Institute of Technology	Harvard University	...	University of Southern California	University of Texas Arlington	University of Texas Austin	University of Texas Dallas
admit															
0	NaN	0.719651	0.758365	0.767644	0.712950	0.831969	0.771135	0.640469	0.788180	0.653408	...	0.690210	0.646859	0.803169	0.718465
1	1.0	0.772896	0.576833	0.754546	0.721456	0.763429	0.812594	0.648206	0.788944	0.786333	...	0.796017	0.687317	0.837892	0.744687

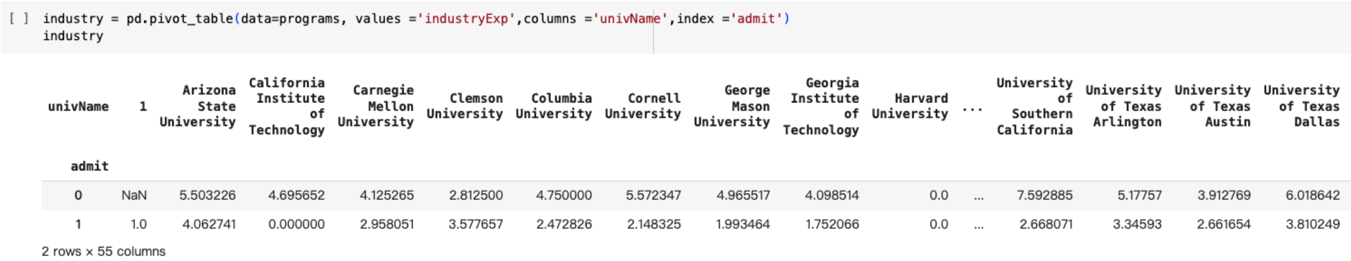
2 rows × 55 columns



```
[ ] research = pd.pivot_table(data=programs, values ='researchExp',columns ='univName',index ='admit')
    research
```

univName	1	Arizona State University	California Institute of Technology	Carnegie Mellon University	Clemson University	Columbia University	Cornell University	George Mason University	Georgia Institute of Technology	Harvard University	...	University of Southern California	University of Texas Arlington	University of Texas Austin	University of Texas Dallas
admit															
0	NaN	0.366667	0.608696	0.292994	0.226974	0.321429	0.472669	0.413793	0.617670	0.230769	...	0.300395	0.158879	0.642225	0.304927
1	1.0	0.500826	0.000000	0.414073	0.376022	0.315217	0.124402	0.205882	0.342975	0.000000	...	0.348778	0.095930	0.939850	0.204986

2 rows × 55 columns



```
[ ] industry = pd.pivot_table(data=programs, values ='industryExp',columns ='univName',index ='admit')
    industry
```

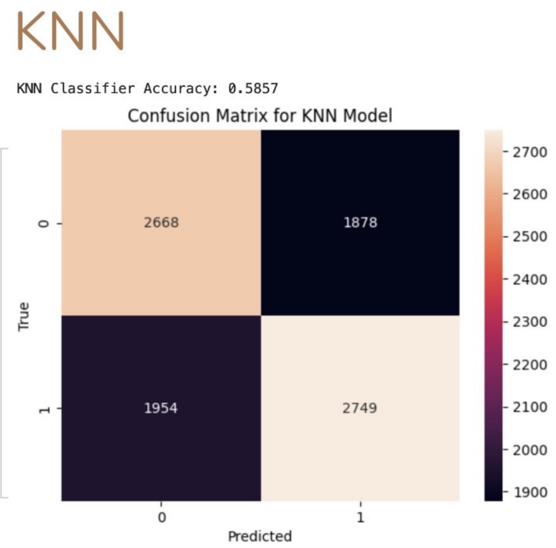
univName	1	Arizona State University	California Institute of Technology	Carnegie Mellon University	Clemson University	Columbia University	Cornell University	George Mason University	Georgia Institute of Technology	Harvard University	...	University of Southern California	University of Texas Arlington	University of Texas Austin	University of Texas Dallas
admit															
0	NaN	5.503226	4.695652	4.125265	2.812500	4.750000	5.572347	4.965517	4.098514	0.0	...	7.592885	5.17757	3.912769	6.018642
1	1.0	4.062741	0.000000	2.958051	3.577657	2.472826	2.148325	1.993464	1.752066	0.0	...	2.668071	3.34593	2.661654	3.810249

2 rows × 55 columns

8 Experiments

The following results were obtained in the previous milestone, so there might be some discrepancy compared to the accuracy obtained from the code, since we made some modification and re-ran the code. The discrepancy is small, within 3 percent, and it does not affect any result in the final paper so we proceed with it.

We trained the K-nearest-neighbor model with training set and use it to predict the test set. The following confusion matrix is the accuracy result for test set:



KNN

```
KNN Classifier Accuracy: 0.5857
```

Then we can calculate the precision: $\frac{2749}{(2749+1954)} = 0.5845$ and the recall: $\frac{2749}{(2749+2668)} = 0.5074$. The precision, recall, and accuracy of the KNN model are poorly performed, with slightly higher than 50 percent accuracy. In our binary classification model, this result has no difference than random guessing so we continue our research with other models.

SVM Test Accuracy for Different Kernels

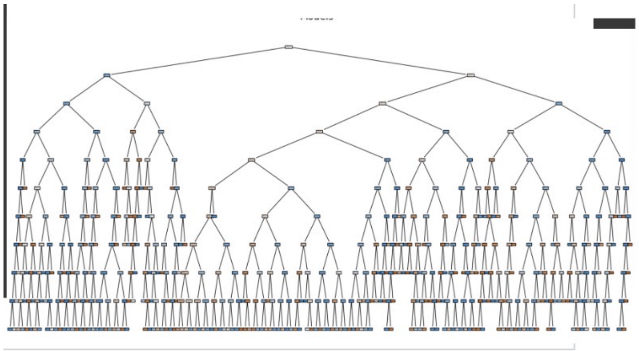
RBF kernel	Polynomial Kernel	Sigmoid Kernel
0.7099	0.7000	0.5659

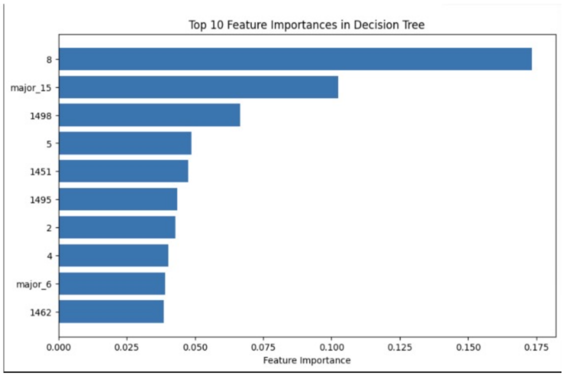
The RBF kernel achieved the highest accuracy among the three, with a score of 0.7099, indicating that it was the most effective in capturing the complexities and patterns in the dataset. It indicates the non-linear pattern of the distribution of the observations in the dataset.

The Polynomial kernel followed closely with an accuracy of 0.7000. This result suggests that the data also exhibits polynomial characteristics, but the degree and nature of these polynomial features might not be as pronounced or as fitting to the model as those captured by the RBF kernel.

The Sigmoid kernel has the lowest accuracy among the three kernels, 0.5659. This suggests that the Sigmoid kernel's transformation of the data was not as suitable for this particular dataset, potentially due to its tendency to transform data in a way that doesn't align well with the underlying patterns needed for accurate classification in this case.

The GridSearch states the best decision tree is the one with hyperparameter of 30 maximum depth and 15 minimum Sample Split. We then draw the decision tree and plot the top 10 importance of the feature:

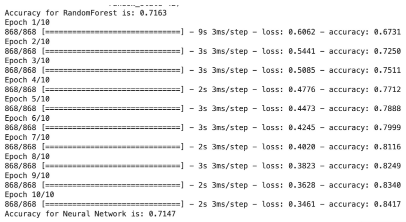




The tree generated is complicated and almost impossible to interpret by reading the graph itself. The 10 most important features of the model are either majors or the program names. This violates our hypothesis that students' academical scores are the most weighted feature in the admission process.

Our Neural Network model includes 3 Dense layers with Relu and sigmoid as activation function, and 1 drop-out layer. We select adam as the optimzer, binary cross-entropy as the loss function, choose a minibatch size of 32, and perform 10 epochs. The following are the details of the model and accuracy across each epoch.

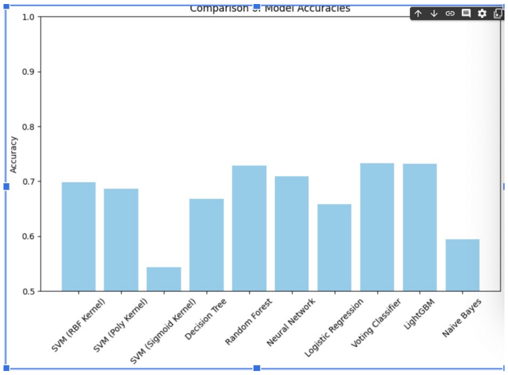
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               224512

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 232833 (909.50 KB)
Trainable params: 232833 (909.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



```
Accuracy for RandomForest is: 0.7163
Epoch 1/10
868/868 [==============================] - 9s 3ms/step - loss: 0.6062 - accuracy: 0.6731
Epoch 2/10
868/868 [==============================] - 3s 3ms/step - loss: 0.5441 - accuracy: 0.7250
Epoch 3/10
868/868 [==============================] - 3s 3ms/step - loss: 0.5085 - accuracy: 0.7511
Epoch 4/10
868/868 [==============================] - 2s 3ms/step - loss: 0.4776 - accuracy: 0.7712
Epoch 5/10
868/868 [==============================] - 3s 3ms/step - loss: 0.4473 - accuracy: 0.7888
Epoch 6/10
868/868 [==============================] - 3s 3ms/step - loss: 0.4245 - accuracy: 0.7999
Epoch 7/10
868/868 [==============================] - 2s 3ms/step - loss: 0.4020 - accuracy: 0.8116
Epoch 8/10
868/868 [==============================] - 3s 3ms/step - loss: 0.3823 - accuracy: 0.8249
Epoch 9/10
868/868 [==============================] - 2s 3ms/step - loss: 0.3628 - accuracy: 0.8340
Epoch 10/10
868/868 [==============================] - 2s 3ms/step - loss: 0.3461 - accuracy: 0.8417
Accuracy for Neural Network is: 0.7147
```

Although we have over 200000 parameters used in the neural network and the accuracy of the model increases from 67 percent to 84 percent on the training set, the final test accuracy is still slightly higher than 70 percent. This might indicate the limitation of our data related to our research question.

9 Results



We collected test accuracy metrics from all models, and their distribution was subsequently visualized via a histogram for enhanced analytical clarity. Amongst the evaluated models, the Random Forest, Voting Classifier, Neural Network, and LightGBM distinguished themselves by achieving the highest levels of accuracy. Notwithstanding their relatively superior performance across the spectrum of models, it is noteworthy that the maximal accuracy plateaued at just above 70 percent. This finding warrants further investigation into potential limitations within the modeling framework or data characteristics. Additionally, a noteworthy observation was made regarding the Neural Network model: an increment in the number of epochs resulted paradoxically in a decrease in model accuracy, indicating the existence of overfitting.

Here are the final Test Accuracies:

'KNN': 0.5857
'SVM (RBF Kernel)': 0.7099
'SVM (Poly Kernel)': 0.7000
'SVM (Sigmoid Kernel)': 0.5659
'Decision Tree': 0.6466
'Random Forest': 0.7285
'Neural Network': 0.7147
'Logistic Regression': 0.6872
'Voting Classifier': 0.7361
'LightGBM': 0.7323
'Naive Bayes': 0.5017

Graduation Recommendation Program Software: Link to Jupyter Notebook

10 Conclusion

The test accuracies of various models we applied do not generate satisfying results for the prediction. According to the feature importance analysis of the Decision Tree, a large proportion of the high-ranking features are school names, which indicates that different schools should have different numeric evaluations of their importance/ranking. However, our data set currently does not reflect this feature. Also, other important factors in the admission decision process such as statement of purpose and interview performance should be taken into consideration in our data set. Given these deficiencies stated above and the room for improvement, our model can achieve an over 0.7 accuracy prediction given program applicants' information and the school they are applying to.

11 Team Contribution

Tianze Ren: data processing, literature review, created slides for the presentation
Kaicheng Chu: model training, accuracy comparison, graph plotting, and slides presentation
Haley Fitch: testing and setting up different models to train the data with and calculate better test accuracy, video editing and creation of the ML4VA video presentation

12 Citations and References

1.https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluatio
2.https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
3.https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
4.https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
5.https://keras.io/examples/
6.https://www.jstor.org/stable/23353972
7.Kern Alexander, Journal of Education Finance, Vol. 38, No. 2 (FALL 2012), pp. 170-176 (7 pages)
8.https://taxfoundation.org/blog/food-thought-information-asymmetries-and-market-higher-education/
9.Liz Malm, Tax Fundation, July 30, 2012
10.https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html
11.https://lightgbm.readthedocs.io/en/latest/Python-Intro.html
12.https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html
13.https://www.geeksforgeeks.org/naive-bayes-classifiers/