Graduate Admission Chance Prediction

Team 211: Weiqiao Shen, Peijia Wang, Anqi Zhao, Siyue Zhu

I. Background

As one of the terrifying viruses in human history, Covid-19 has invaded the whole world and changed people's lives. Because of social distancing, many things are suspended or disrupted. For example, schools turn to online teaching and people have to work remotely. Unsurprisingly, as the standard test for an admission requirement, GRE offers an online version in the United States. At the same time, many colleges removed the admission requirements for GRE scores. These phenomena trigger our thinking: What factors influence college admission and how do they affect it? To find the answer, we analyze the pattern behind the data and use three classification machine models to help predict the admission result. In the following sections, we'll uncover the mystery of how colleges measure whether an applicant is eligible for admission.

II. Guide Questions

What are the factors that influence college admission and how do they affect it? We focus on factors that may affect the college admission result, including GRE and TOEFL scores but not limited to them. There are many other factors such as research experience, statement of purpose (SOP), and letter of recommendation (LOR). These factors are encoded into scales of 0-5 or 0-10 in our data set, which is analytic friendly.

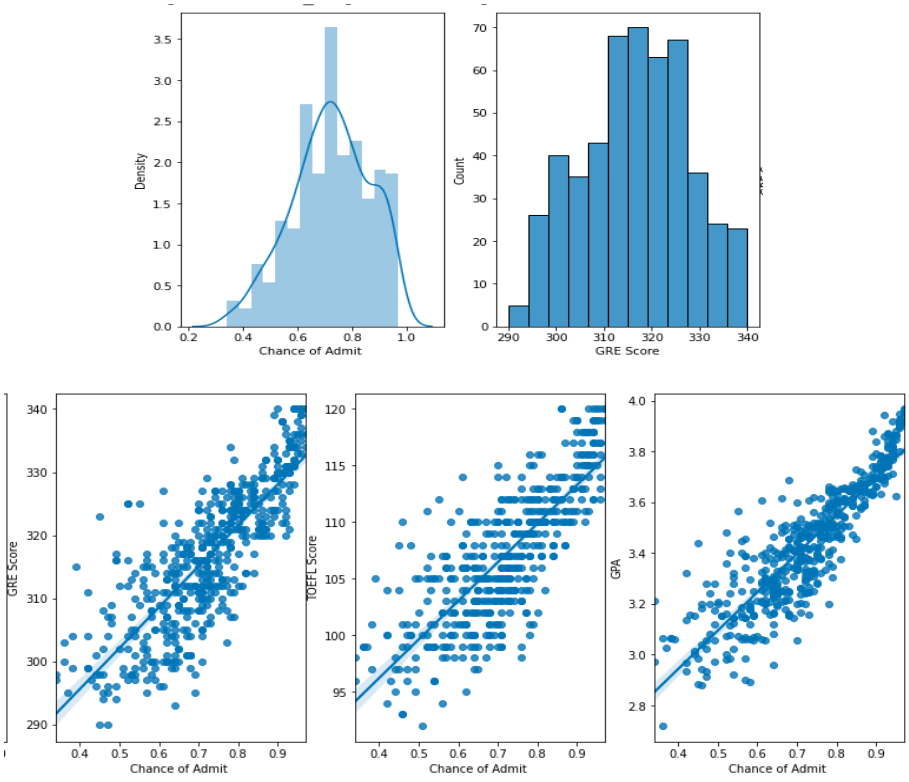
III. Data Wrangling

We first imported data into our notebook and then checked if there are any null values. Fortunately, the dataset is very clean and doesn't have any missing value. We noticed the variable "chance of admit" is float and couldn't be fed into the model directly, so we encoded the original variable into a new nominal variable called "admit" as the output of our model. We generated it by assigning the "chance of admit", a numerical variable in the range of 0 to 1, into four bins. Since we didn't have any value smaller than 0.2, we define the probability of being admitted of 0.2 - 0.4 as "unlikely," 0.4 - 0.6 as "possible," 0.6 - 0.8 as "likely," and >0.8 as "certain". We also converted the variable "CGPA," which represents the cumulative grade point average ranging from 0 to 10, to "GPA", which scales from 0 to 4, because GPA is more widely used.

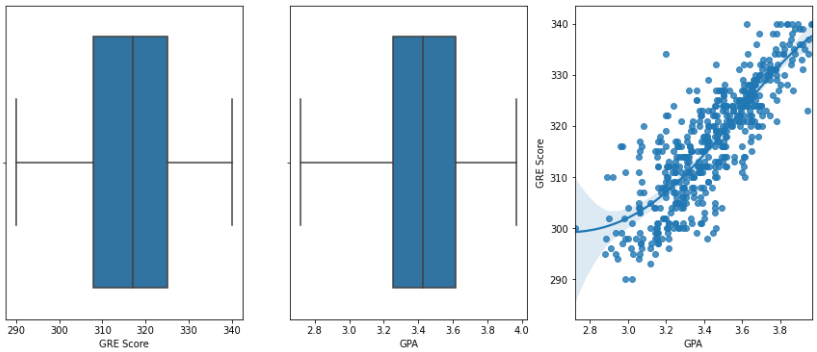
	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit	Admit	GPA
0	1	337	118	4	4.5	4.5	9.65	1	0.92	certain	3.860
1	2	324	107	4	4.0	4.5	8.87	1	0.76	likely	3.548
2	3	316	104	3	3.0	3.5	8.00	1	0.72	likely	3.200
3	4	322	110	3	3.5	2.5	8.67	1	0.80	certain	3.468
4	5	314	103	2	2.0	3.0	8.21	0	0.65	likely	3.284

IV. Exploratory Data Analysis

We found the distribution of the chance of admitting and GRE scores are both in a skewed bell shape with the median around 0.7 and 315, respectively. The correlation between the possibility of admitting and factors, including GRE score, TOEFL score, and GPA, are more or less positive.

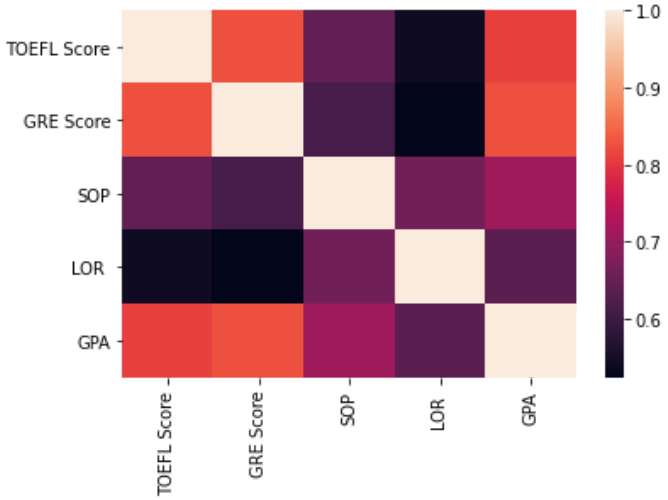


Then we plotted the GRE score and GPA and explored the correlation between these two factors. We discovered that the median score for GRE is around 317, and the median GPA is 3.4. As GRE and GPA are both positively correlated with the chance of admitting, we are curious about the correlation between GRE and GPA. According to the third graph, GRE score and GPA have a positive linear relationship. Therefore, we speculate that students who get a high GRE score are likely to have a high GPA.



To further explore the relationship between features, we used a heatmap, where the lighter color represents a strong correlation, to display the correlation for five factors together. The most outstanding information we can read from the graph is that the letter of recommendation has a very weak or even correlation with other elements. Besides, GPA is highly related to TOEFL score and GRE score, which implies that applicants who perform well academically can also do well on TOEFL and GRE.

	TOEFL Score	GRE Score	SOP	LOR	GPA
TOEFL Score	1.000000	0.827200	0.644410	0.541563	0.810574
GRE Score	0.827200	1.000000	0.613498	0.524679	0.825878
SOP	0.644410	0.613498	1.000000	0.663707	0.712154
LOR	0.541563	0.524679	0.663707	1.000000	0.637469
GPA	0.810574	0.825878	0.712154	0.637469	1.000000



V. Data Modeling

We applied different models to our data and tried to see which one best fits. These models include logistic regression, k nearest neighbor, and random forest. We have tried SVM, but we failed to get good feedback. Thus we will not discuss SVM in this report. The features selected are “GRE Score,” “TOEFL Score,” “University Rating,” “SOP,” “Research,” and “GPA.” And the output, also known as the target, is “Admit.” Besides, we also use GridSearchCV to tune the hyperparameter when training K Nearest Neighbor and Random Forest.

```
features = ['GRE Score',
 'TOEFL Score',
 'University Rating',
 'SOP',
 'Research',
 'GPA']
target = "Admit"
```

```
[ ] X = university[features]
    y = university[target]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=17)
```

1. Logistic Regression

For classification problems, logistic regression is the most classic and useful model because of the property of sigmoid function, so we first adopt this statistical model.

The trained logistic regression gets an accuracy score of 0.81. We also compute the coefficient of each variable. We find that GPA is most strongly correlated with the possibility of getting admission, with coefficients of approximately 2.3. Having research experience is the second important factor that increases the admission probability.

```
model = LogisticRegression(solver='newton-cg')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy_score(y_test, y_pred)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/optimize.py:212: ConvergenceWarning: newton-cg failed to converge. Increase the number of iterations.
  "number of iterations.", ConvergenceWarning)
0.81
```

```
for i in range(len(features)):
  print(features[i] + ": " + str(model.coef_[0][i]))
```

```
GRE Score: 0.09616770006577893
TOEFL Score: 0.27150262195790387
University Rating: 0.41684715030877667
SOP: 0.4544244367932193
Research: 1.16461373599887645
GPA: 2.2955470170313537
```

2. K Nearest Neighbor

K nearest Neighbor is a non-parametric classification model that we used. Non-parametric model means that the method does not presume any mapping function of the training data, but rather it just assumes a pattern that is close to the output variable. KNN model uses data and classifies new data points based on similarity measures (e.g., distance function). Classification is done by a majority vote to its neighbors.

```
[ ] model = KNeighborsClassifier(n_neighbors=15, p=1, weights='distance')
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy_score(y_test, y_pred)
```

```
    0.78
```

```
[ ] parameters = {'n_neighbors':[5,10,15,20,25], 'weights':['uniform','distance'],
                  'algorithm':['auto','ball_tree','kd_tree','brute'], 'p':[1,2]}
    knn = KNeighborsClassifier()
    clf = GridSearchCV(estimator=knn, param_grid=parameters, n_jobs=-1, cv=10)
    clf.fit(X_train, y_train)

    clf.best_params_
```

3. Random Forest

Random Forest is an algorithm that consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest generates a class prediction and the class with the most votes becomes our model's prediction. It functions like a black box, so it is hard to explain how it makes choices and gets the final result. But this model does help to reduce overfitting in decision trees and improve the accuracy.

```
[ ] model = RandomForestClassifier(max_depth=5, n_estimators=100)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy_score(y_test, y_pred)

    0.83

[ ] cross_val_score(model, X_train, y_train, cv=5)

    /usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_split.py:667: UserWarning:
      % (min_groups, self.n_splits)), UserWarning)
    array([0.8125, 0.8   , 0.825 , 0.7375, 0.775 ])

    parameters = {'n_estimators':[5,25,50,100,150,200,250], 'criterion':['gini','entropy'],
                  'max_depth':[5,10,15,20,25], 'max_features':['auto','sqrt','log2']}
    forest = RandomForestClassifier(max_depth=5, max_features='sqrt')
    clf = GridSearchCV(estimator=forest, param_grid=parameters, n_jobs=-1, cv=10)
    clf.fit(X_train, y_train)

    clf.best_params_
```

VI. Interactive Input

Because of the time limit, we do not have enough time to build a well-functioned webpage for our model. Instead, we add an interactive input in python using ipywidgets so that applicants can type in their information and use our model to predict the likelihood of getting admitted. It requires providing information on one's GRE score, TOEFL score, the university rating they are applying for, the quality of their statement of purpose, whether they have research experience or not, and their GPA score. With this information, our model will print out the prediction of their probability of admission.

```
gre = input('Enter your GRE score: ')
toefl = input('Enter your TOEFL score: ')
rating = input('Enter the rating of the university you apply for: ')
import ipywidgets as iw
from IPython.display import display
sop=iw.Dropdown(options=[0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5], value=0, description='SOP: ')
print('Enter your Statement of Purpose score: ')
display(sop)
print('Do you have any research experience?')
research = iw.Dropdown(options=[0,1], value=0, description='research: ')
display(research)
gpa = input('Enter your gpa: ')

...  Enter your GRE score: 330
     Enter your TOEFL score: 110
     Enter the rating of the university you apply for: 4
     Enter your Statement of Purpose score:
           SOP: 0
     Do you have any research experience?
        research: 0
     Enter your gpa:
```

VII. Conclusions

According to our analysis, test scores like GPA, GRE, and TOEFL are all highly correlated to the chance of getting admitted, especially for GPA, which has a strong positive correlation with the chance of getting admitted. It is interesting that, in real practice, TOFEL is not taken for assessment in graduate admission. In other words, TOFEL may imply applicants' academic performance and help to predict one's admission result, but it does not mean it matters in graduate admission. Applicants can get a probability of getting admitted to their dream school simply by inputting their basic information into the model. With the outcome generated by the model, one could list out the university they could apply for dependent on the probability and won't lose their chance of getting into a better university.

However, there are several drawbacks of our model:

1. The dataset limits our model. The dataset we use does not contain all relevant features and covers only a tiny portion of all situations.
2. Since the data size is not large, we have to limit the category of our model's outcome. So far, we only have four distinct outcomes that only tell people how likely they will be admitted without a precise numeric answer.
3. The model is limited to international students who have TOEFL scores.

Overall speaking, the model works well for the students in this given dataset. In the future, if we obtain more high-quality and comprehensive data, we might be able to train our model better or construct advanced models to generate a more diverse and precise outcome. By utilizing the model with best performance, it can be used in a widespread range after we design a webpage combining frontend and backend knowledge.

GitHub Link: https://github.com/angelazhao2552/stanford-datathon-team211