Probability of entering your dream Graduation Program

Introduction

Whether one can enter a graduate program is a mystery; Sometimes, your classmates who seem not as good as you may be admitted to a program you dream while sometimes you can be admitted to programs that you think you must be denied. In this case, I want to have a logical prediction of whether one can enter their dream program, what is the probability of entering it, and, more importantly, what one can do to improve their probability of being admitted. It is more reliable than some admission consulting companies that give decisions just based on their experience. My target variable is the Chance of Admitted, a regression variable because it indicates probability. After dropping the "Serial No." which is not meaningful, there are 3200 data points, containing 400 rows with 8 columns, with 7 features and 1 target variable (figure 1).

In all features, GRE Scores, TOEFL Scores, and Undergraduate GPA are continuous variables because they are random numbers in a range, having clear minimum and maximum; University Rating (1-5), Statement of Purpose (1.0 – 5.0), and Letter of Recommendation Strength (1.0 – 5.0) are Ordinary variables because they are all in ranked, while having different levels; Research Experience is categorized variable because the result is 0 or 1 indicating yes or no. This data set is well documented, I found this in Kaggle, from the UCLA database. Some people use it as basic EDA, and some people use it to predict the result of admission when inputting certain data of each feature. Most of them use a linear regression model.

Exploratory Data Analysis

First, I visualized the target variables (figure 2). I find that there are not many outliers in Chance of Admit. The mean of it is 0.724 (72.4%), higher than I expected. The standard deviation is 0.1426, and all data points are in the range of 0.34 – 0.97.

Then, I did a heatmap to show a broad correlation between the Chance of Admission, my target variable, and all other features (figure 3). Closer to 1, the more relative they are. Not surprisingly, GRE, Tofel, and GPA are three figures that have the highest correlation to the target variable, around 8, and the lowest score is research, 0.55.

After finding GPA and GRE are two of the most correlated variables to my target, I did a scalar plot of both to the Chance of Admit because they are continuous variables (figure 4 and 5). I found that they are both kindly in linear correction to the Chance of admission. The slope of CGPA is nearly 45 degrees, while the GRE one has a shallower slope here. That means an increase of the same percentage of CGPA will affect the target more than GRE did.

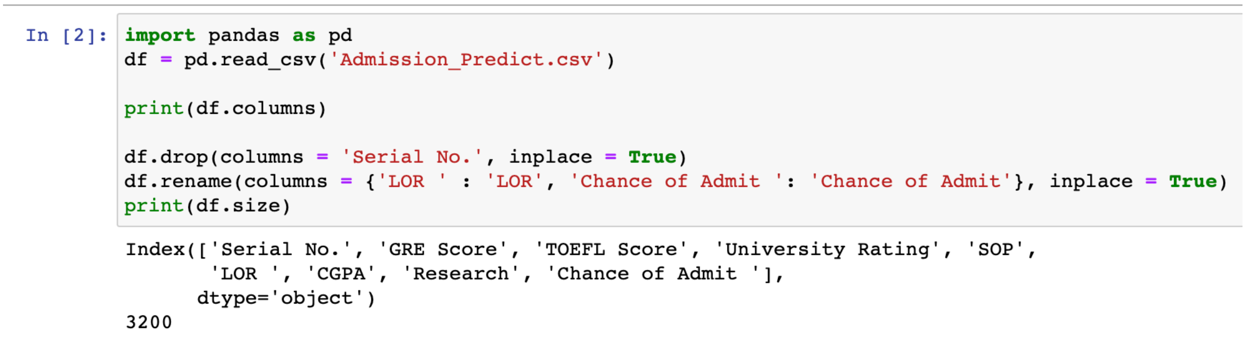
For ordinary variables, I use a boxplot graph because a bar chart is not good with many categories (figures 6 and 7). For a LOR, I found that increasing LOR will effetely increase the mean of Chance of Admit. However, there is a huge increase in the mean when we increase LOR from 4.0 to 4.5, while there isn't much change when we increase LOR from 4.5 to 5.0. That means that after a certain level of strength of the recommendation letter, it will be less effective. Also, If we want a chance of admission over 90%, we need a LOR strength of more than 3. For SOP, it is quite similar to LOR. But the biggest difference there is that it has more outliers. Compared to LOR, which nearly has no outlier, even if you have a maximum of 5.0 in SOP, there is one data point that has a chance of being admitted lower than 40%. It also happened when SOP equals 2.0, 3.0, and 4.0.

For categorized variables, the change in admitted rate based on whether or not we have research (figure 8), I found two facts. One is that most people who do not have research have less than a 75% chance of admission, and the second is that if you want a chance of admission over 90%, you need to have research.

Data preprocessing

I just use a simple split in my data set because the target variables are not away from equally distributed. I initially wanted to use stratified splitting based on University rating, but because it is a feature instead of target variables, I listened to the comments in the presentation and changed it to a simple split. My data set is IID because there are no group features inside, and it is not time-series data. Because it is a well-organized and designed dataset, there are no missing values inside (figure 9), and the data inputs are all integers instead of strings. It uses 0 and 1 to represent yes or no in research and has a clear ranking of 1-0 in ordinary features. What I only need to do is to standardize continuous valuables, including GRE, CGPA, and Tofel. I use a min-max scaler here because it has clear ranges in all of them, and they all transfer into 0 – 1 in the end. (figure 10)

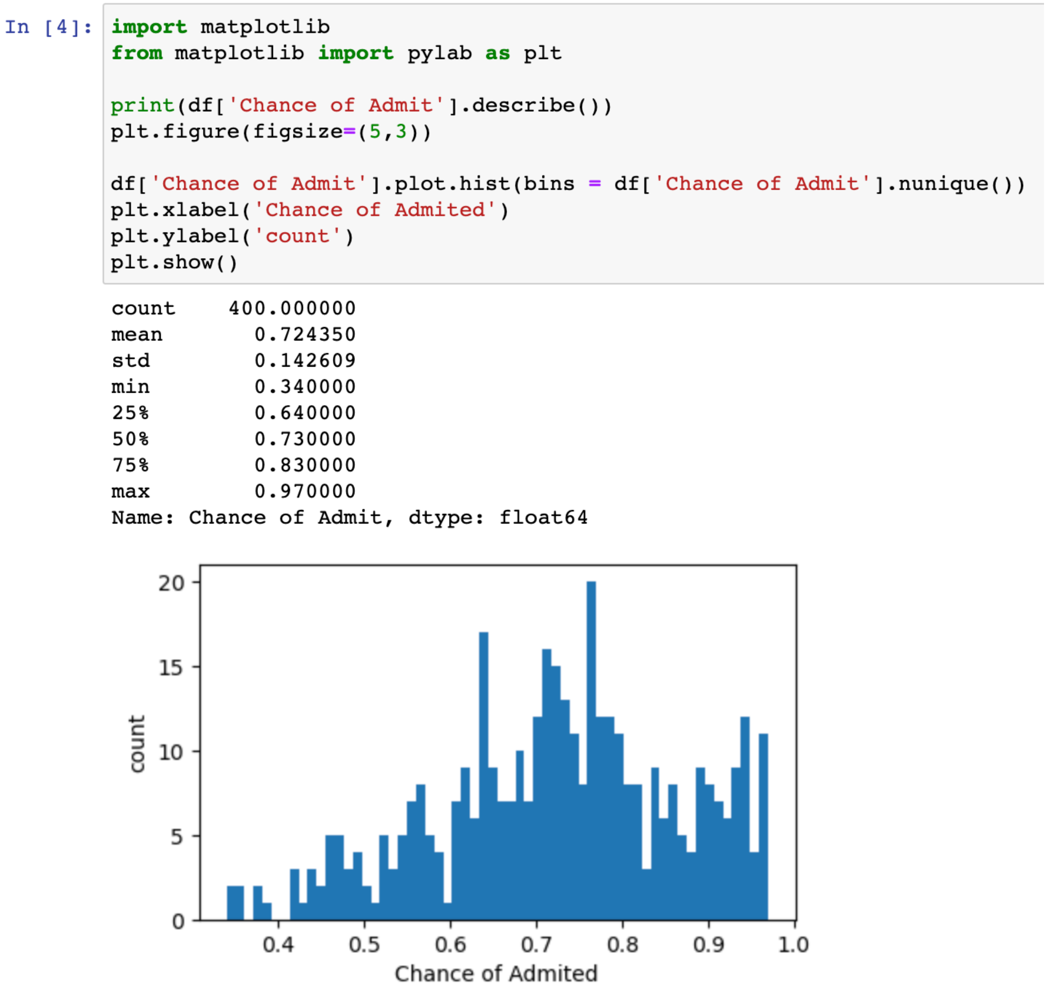
Figure 1



```
In [2]: import pandas as pd
        df = pd.read_csv('Admission_Predict.csv')

        print(df.columns)

        df.drop(columns = 'Serial No.', inplace = True)
        df.rename(columns = {'LOR ' : 'LOR', 'Chance of Admit ': 'Chance of Admit'}, inplace = True)
        print(df.size)
```

```
Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating', 'SOP',
       'LOR ', 'CGPA', 'Research', 'Chance of Admit '],
      dtype='object')
3200
```

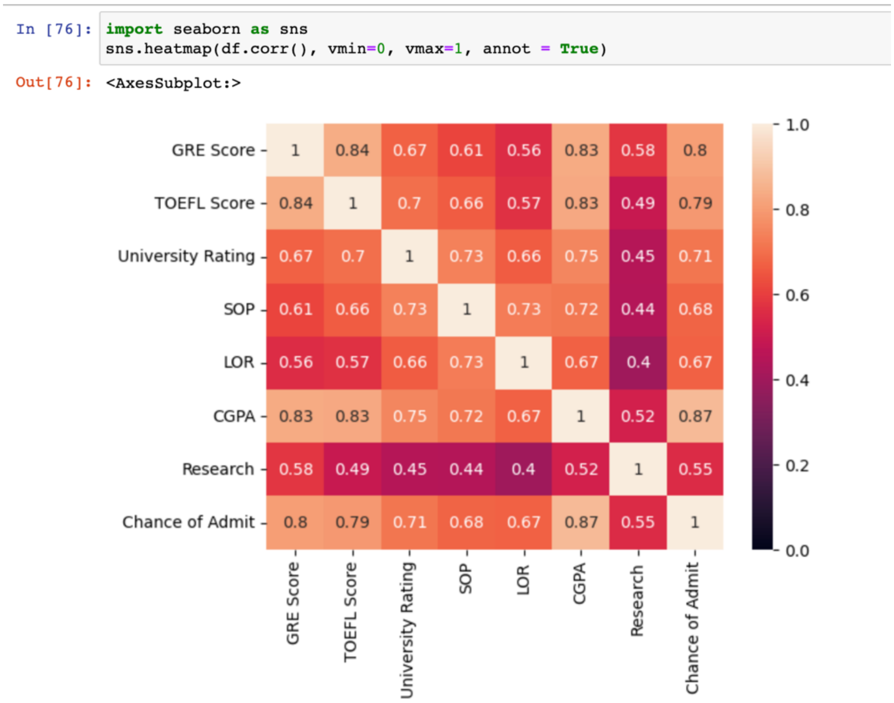
Figure 2



```
In [4]: import matplotlib
        from matplotlib import pylab as plt

        print(df['Chance of Admit'].describe())
        plt.figure(figsize=(5,3))

        df['Chance of Admit'].plot.hist(bins = df['Chance of Admit'].nunique())
        plt.xlabel('Chance of Admited')
        plt.ylabel('count')
        plt.show()
```

```
count    400.000000
mean       0.724350
std        0.142609
min        0.340000
25%        0.640000
50%        0.730000
75%        0.830000
max        0.970000
Name: Chance of Admit, dtype: float64
```

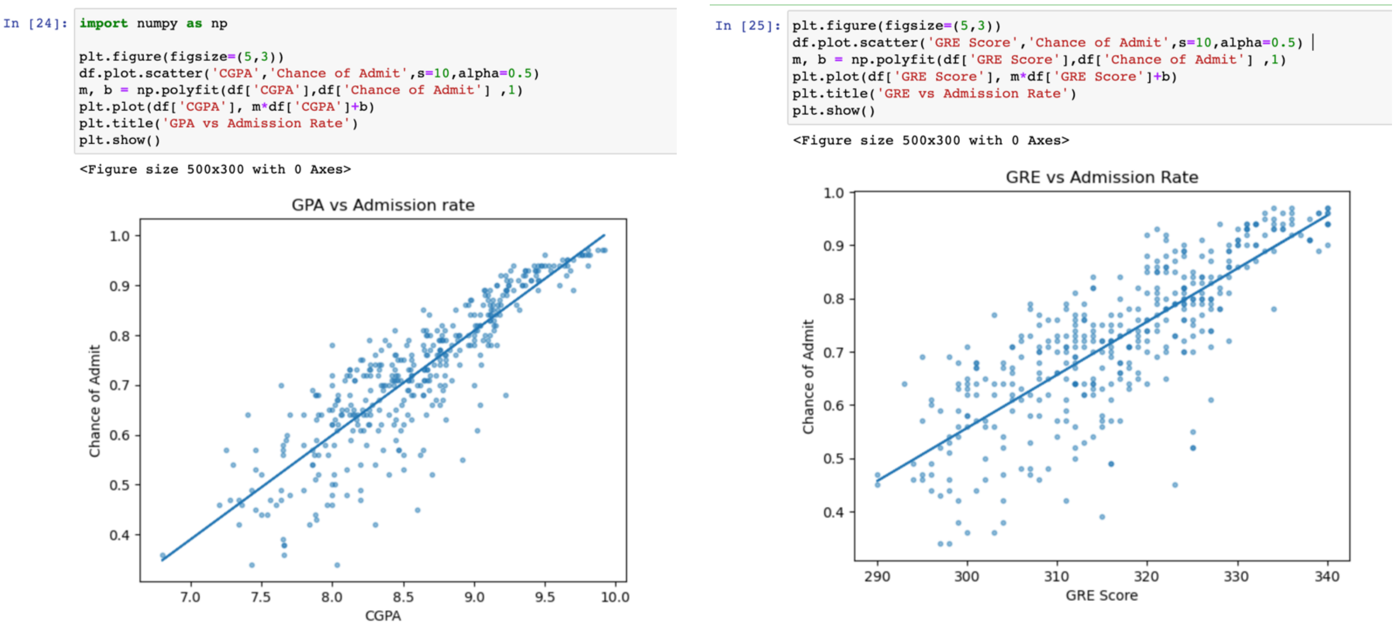
Figure 3



```
In [76]: import seaborn as sns
         sns.heatmap(df.corr(), vmin=0, vmax=1, annot = True)

Out[76]: <AxesSubplot:>
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
GRE Score	1	0.84	0.67	0.61	0.56	0.83	0.58	0.8
TOEFL Score	0.84	1	0.7	0.66	0.57	0.83	0.49	0.79
University Rating	0.67	0.7	1	0.73	0.66	0.75	0.45	0.71
SOP	0.61	0.66	0.73	1	0.73	0.72	0.44	0.68
LOR	0.56	0.57	0.66	0.73	1	0.67	0.4	0.67
CGPA	0.83	0.83	0.75	0.72	0.67	1	0.52	0.87
Research	0.58	0.49	0.45	0.44	0.4	0.52	1	0.55
Chance of Admit	0.8	0.79	0.71	0.68	0.67	0.87	0.55	1

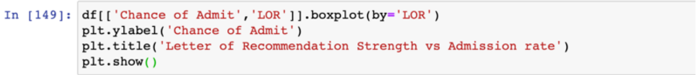
Figure 4 & 5



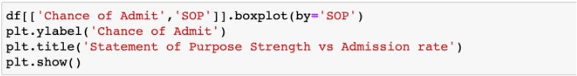
```
In [24]: import numpy as np

         plt.figure(figsize=(5,3))
         df.plot.scatter('CGPA','Chance of Admit',s=10,alpha=0.5)
         m, b = np.polyfit(df['CGPA'],df['Chance of Admit'] ,1)
         plt.plot(df['CGPA'], m*df['CGPA']+b)
         plt.title('GPA vs Admission rate')
         plt.show()

         <Figure size 500x300 with 0 Axes>
```

```
In [25]: plt.figure(figsize=(5,3))
         df.plot.scatter('GRE Score','Chance of Admit',s=10,alpha=0.5)
         m, b = np.polyfit(df['GRE Score'],df['Chance of Admit'] ,1)
         plt.plot(df['GRE Score'], m*df['GRE Score']+b)
         plt.title('GRE vs Admission Rate')
         plt.show()

         <Figure size 500x300 with 0 Axes>
```

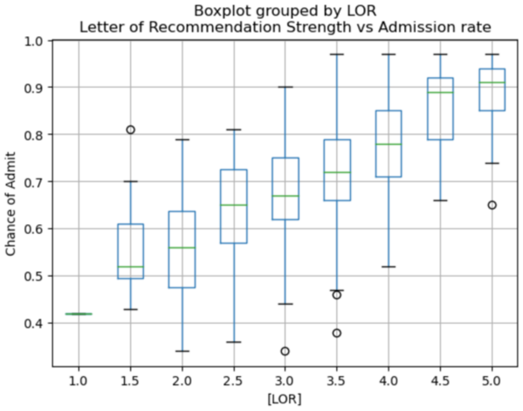
Figure 6 & 7



```
In [149]: df[['Chance of Admit','LOR']].boxplot(by='LOR')
          plt.ylabel('Chance of Admit')
          plt.title('Letter of Recommendation Strength vs Admission rate')
          plt.show()
```



```
df[['Chance of Admit','SOP']].boxplot(by='SOP')
plt.ylabel('Chance of Admit')
plt.title('Statement of Purpose Strength vs Admission rate')
plt.show()
```



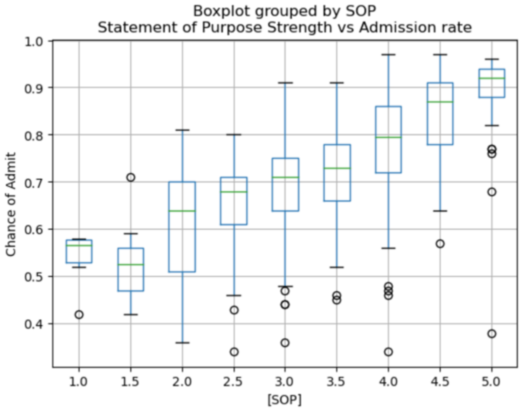
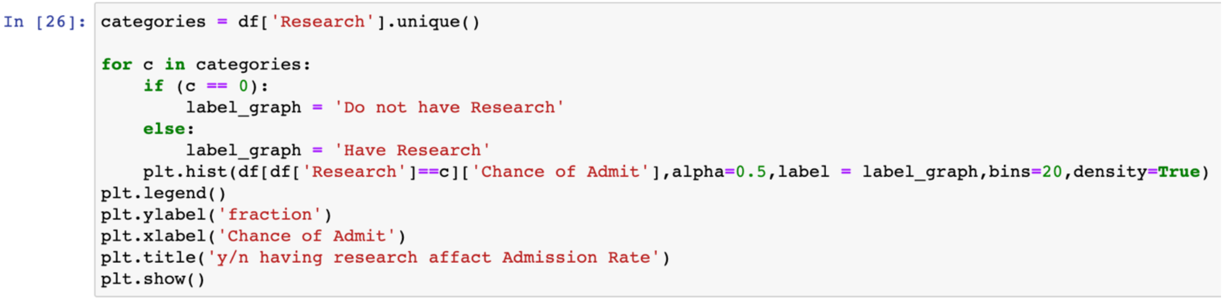


Figure 8



```
In [26]: categories = df['Research'].unique()

         for c in categories:
             if (c == 0):
                 label_graph = 'Do not have Research'
             else:
                 label_graph = 'Have Research'
             plt.hist(df[df['Research']==c]['Chance of Admit'],alpha=0.5,label = label_graph,bins=20,density=True)
         plt.legend()
         plt.ylabel('fraction')
         plt.xlabel('Chance of Admit')
         plt.title('y/n having research affact Admission Rate')
         plt.show()
```

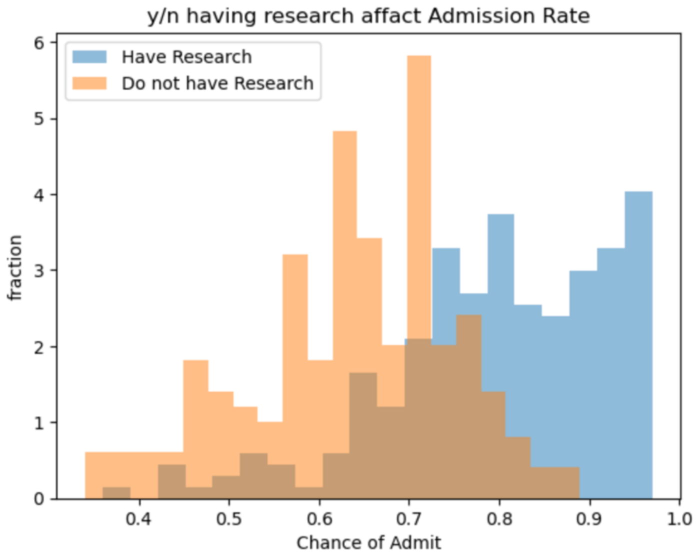
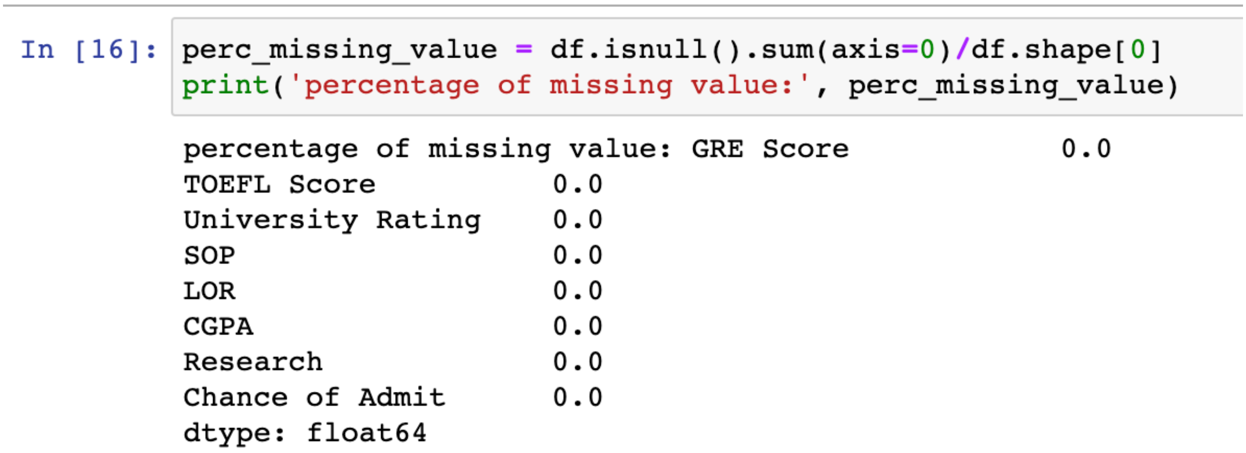


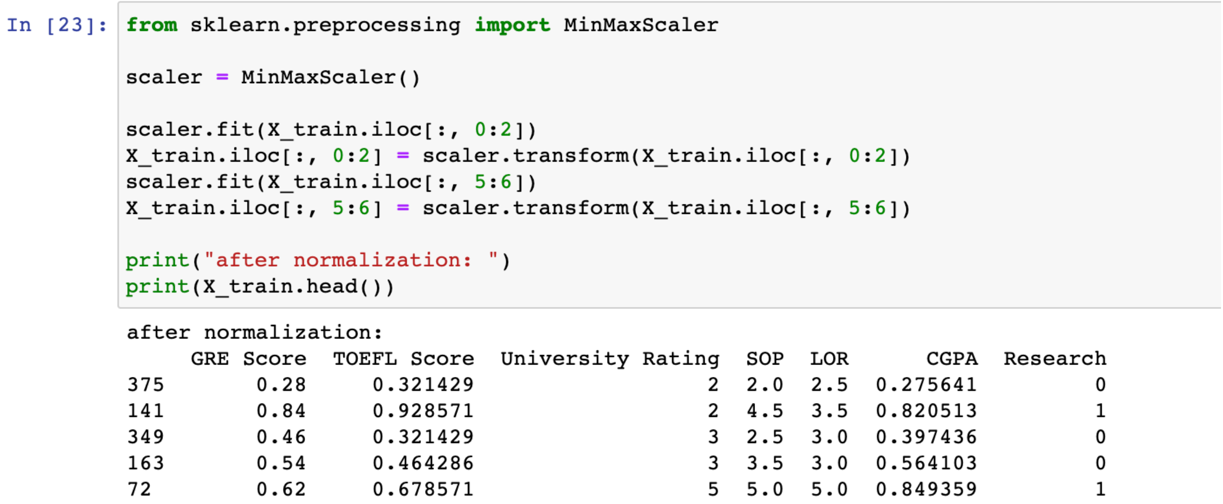
Figure 9



```
In [16]: perc_missing_value = df.isnull().sum(axis=0)/df.shape[0]
         print('percentage of missing value:', perc_missing_value)
```

```
percentage of missing value: GRE Score            0.0
TOEFL Score          0.0
University Rating    0.0
SOP                  0.0
LOR                  0.0
CGPA                 0.0
Research             0.0
Chance of Admit      0.0
dtype: float64
```

Figure 10



```
In [23]: from sklearn.preprocessing import MinMaxScaler

         scaler = MinMaxScaler()

         scaler.fit(X_train.iloc[:, 0:2])
         X_train.iloc[:, 0:2] = scaler.transform(X_train.iloc[:, 0:2])
         scaler.fit(X_train.iloc[:, 5:6])
         X_train.iloc[:, 5:6] = scaler.transform(X_train.iloc[:, 5:6])

         print("after normalization: ")
         print(X_train.head())
```

```
after normalization:
     GRE Score  TOEFL Score  University Rating  SOP  LOR      CGPA  Research
375       0.28     0.321429                  2  2.0  2.5  0.275641         0
141       0.84     0.928571                  2  4.5  3.5  0.820513         1
349       0.46     0.321429                  3  2.5  3.0  0.397436         0
163       0.54     0.464286                  3  3.5  3.0  0.564103         0
72        0.62     0.678571                  5  5.0  5.0  0.849359         1
```

Reference

Acharya, Mohan S. “Graduate Admission 2.” *Kaggle*, 28 Dec. 2018, https://www.kaggle.com/datasets/mohansacharya/graduate-admissions?datasetId=14872&sortBy=commentCount.

sreshta140. “Chances of Getting into My Dream University.” *Kaggle*, Kaggle, 14 June 2020, https://www.kaggle.com/code/sreshta140/chances-of-getting-into-my-dream-university.

My own problem set 1-5

Github repository

https://github.com/seanxxy0528/Data-project.git