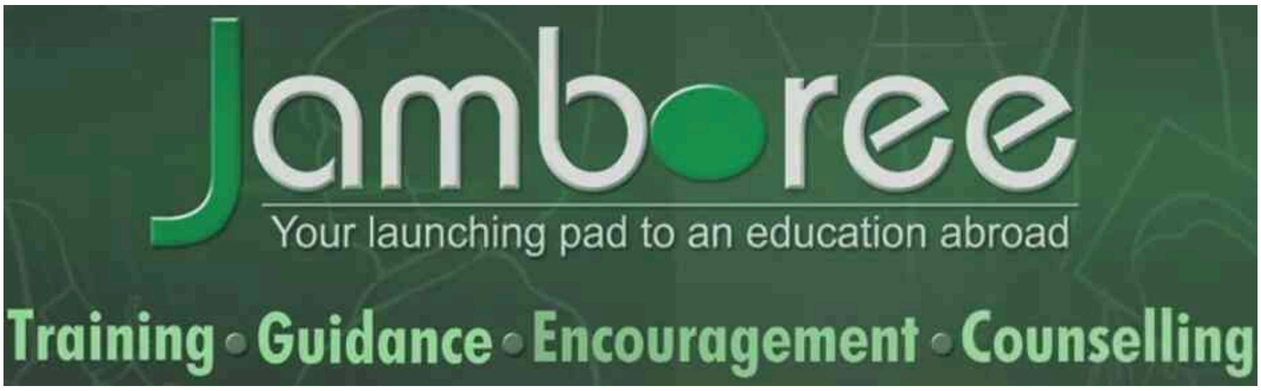

Jamboree
Your launching pad to an education abroad
Training • Guidance • Encouragement • Counselling

About Jamboree

Jamboree has helped thousands of students like you make it to top colleges abroad. Be it GMAT, GRE or SAT, their unique problem-solving methods ensure maximum scores with minimum effort. They recently launched a feature where students/learners can come to their website and check their probability of getting into the IVY league college. This feature estimates the chances of graduate admission from an Indian perspective.

PROBLEM STATEMENT

We are here to make an analysis that will help Jamboree in understanding what factors are important in graduate admissions and how these factors are interrelated among themselves. It will also help predict one's chances of admission given the rest of the variables.

We are going to make an analysis by performing following tasks:

1. Exploratory Data Analysis:

- Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category', missing value detection, statistical summary.
- Univariate Analysis (distribution plots of all the continuous variable(s) barplots/countplots of all the categorical variables)
- Bivariate Analysis (Relationships between important variables such as workday and count, season and count, weather and count).
- Illustrate the insights based on EDA
- Comments on range of attributes, outliers of various attributes
- Comments on the distribution of the variables and relationship between them
- Comments for each univariate and bivariate plots

2. Data Preprocessing:

- Duplicate value check
- Missing value treatment
- Outlier treatment
- Feature engineering
- Data preparation for modeling

3. Model building:

- Build the Linear Regression model and comment on the model statistics
- Display model coefficients with column names
- Try out Ridge and Lasso regression

4. Testing the assumptions of the linear regression model:

- Multicollinearity check by VIF score (variables are dropped one-by-one till none has VIF>5)
- The mean of residuals is nearly zero
- Linearity of variables (no pattern in the residual plot)
- Test for Homoscedasticity
- Normality of residuals (almost bell-shaped curve in residuals distribution, points in QQ plot are almost all on the line)

5. Model performance evaluation:

- Metrics checked - MAE, RMSE, R2, Adj R2
- Train and test performances are checked
- Comments on the performance measures and if there is any need to improve the model or not

6. Actionable Insights & Recommendations:

- Comments on significance of predictor variables
- Comments on additional data sources for model improvement, model implementation in real world, potential business benefits from improving the model

```
# importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#loading the dataset
df=pd.read_csv('/content/Jamboree_Admission.csv')
df.head()               #top five rows of the dataset
```

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

```
df.shape     #finding the rows and columns
```

```
(500, 9)
```

```
df.info()         #displaying information of each column
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Serial No.         500 non-null    int64  
 1   GRE Score          500 non-null    int64  
 2   TOEFL Score        500 non-null    int64  
 3   University Rating  500 non-null    int64  
 4   SOP                500 non-null    float64
 5   LOR                500 non-null    float64
 6   CGPA               500 non-null    float64
 7   Research           500 non-null    int64  
 8   Chance of Admit    500 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 35.3 KB
```

So by looking above,we can say all the columns are numerical. Also there are no coulumns which contains single null values. But still we will cross check for null values

```
df.isnull().sum()           #cheking for null values
```

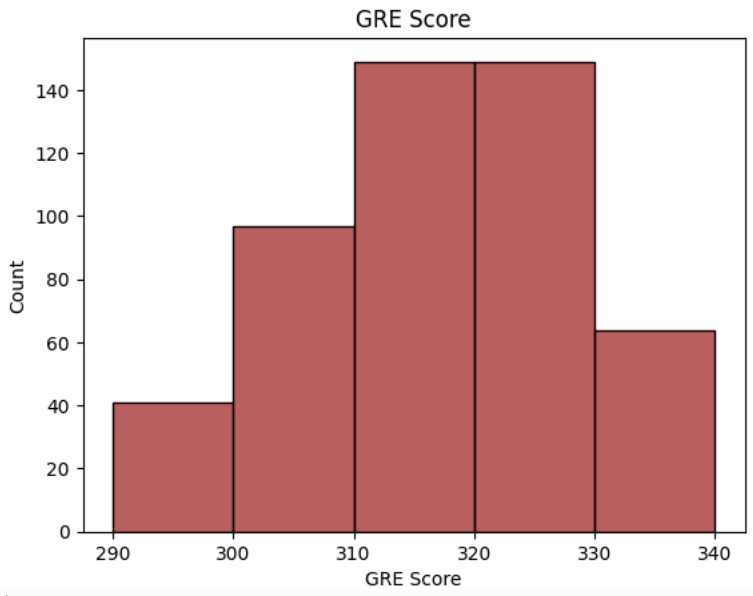
	0
Serial No.	0
GRE Score	0
TOEFL Score	0
University Rating	0
SOP	0
LOR	0
CGPA	0
Research	0
Chance of Admit	0

dtype: int64

Clearly all the columns are not having any null values, which is a good thing about data. Also we can comment that the quality of the data is good

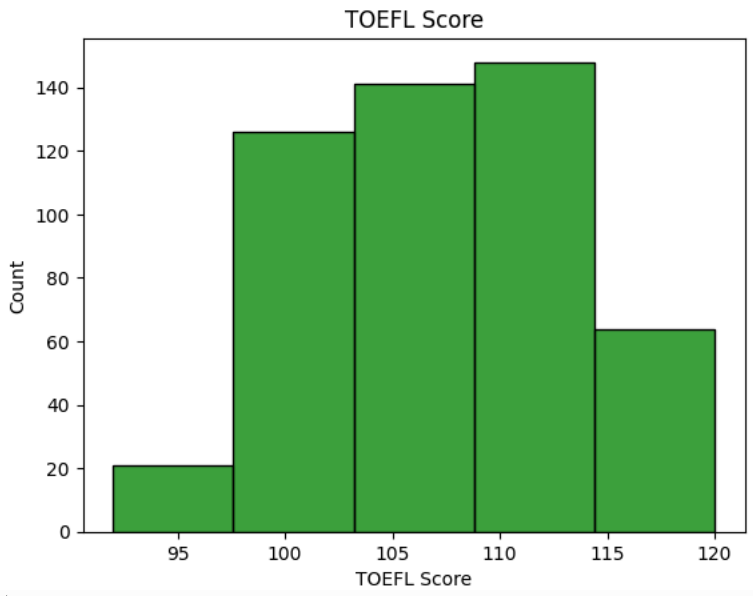
UNIVARIATE ANALYSIS

```
sns.histplot(df['GRE Score'],bins=5,color='brown')
plt.title('GRE Score')
plt.show()
```



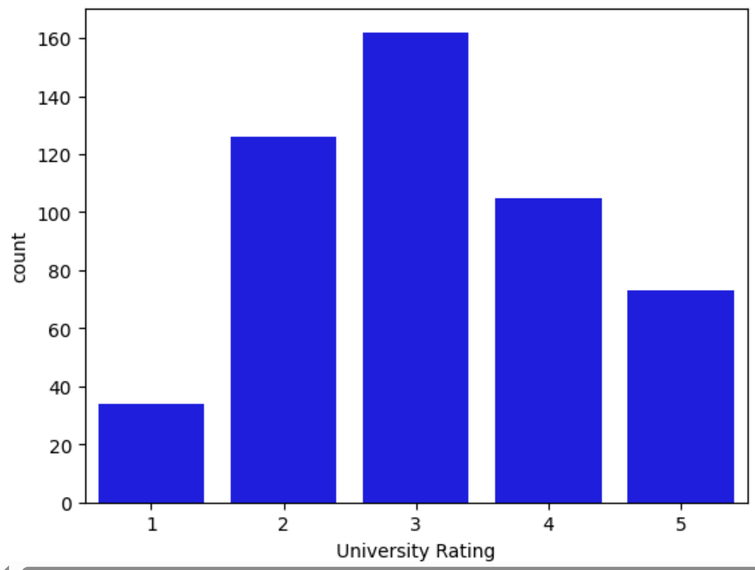
By looking at the plot of GRE Score it can be clearly said that most of the students have the GRE score between 300 and 330

```
sns.histplot(df['TOEFL Score'],bins=5,color='g')
plt.title('TOEFL Score')
plt.show()
```



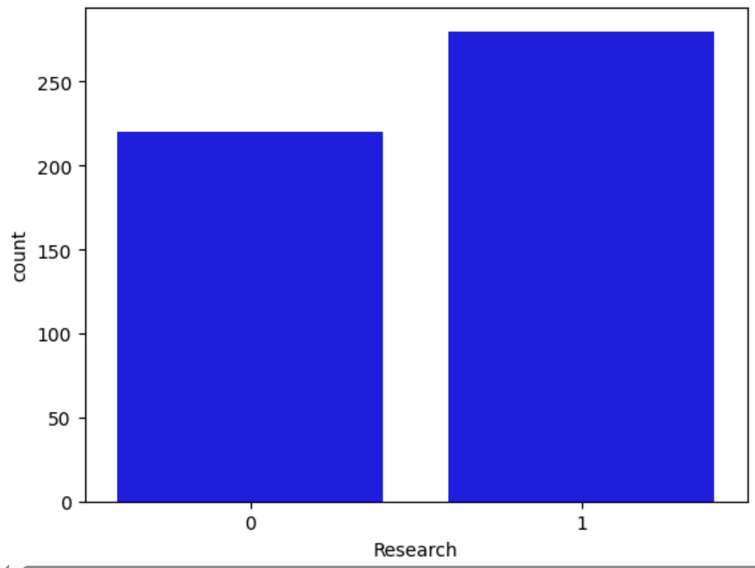
By looking at the countplot of TOEFL Score,we can say that most students have TOEFL Score between 100 and 115

```
sns.countplot(x='University Rating',data=df,color='b')
plt.show()
```



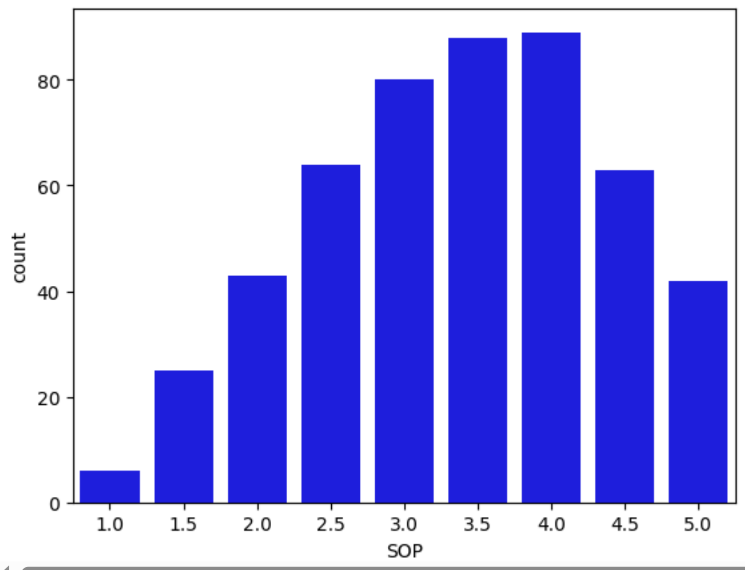
By looking at the countplot of University rating,we can say that most students have rating 3

```
sns.countplot(x='Research',data=df,color='b')
plt.show()
```

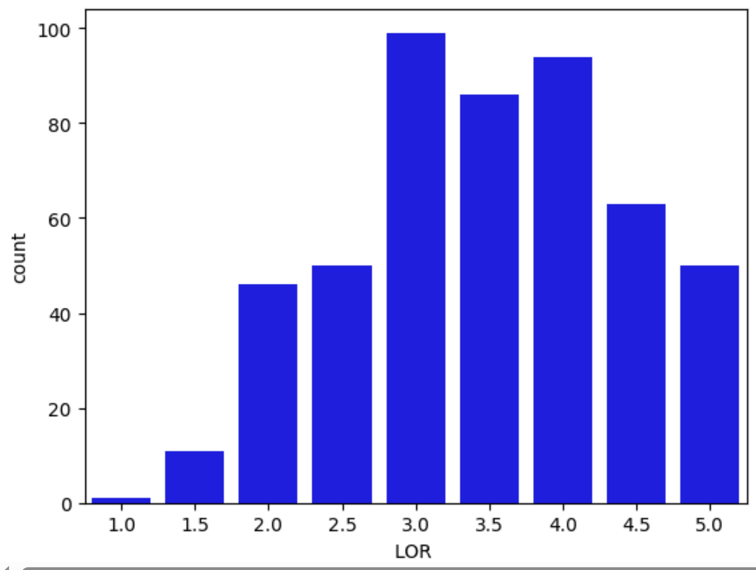


It can be said that although the number of students who have done any research is more but the difference is not big.

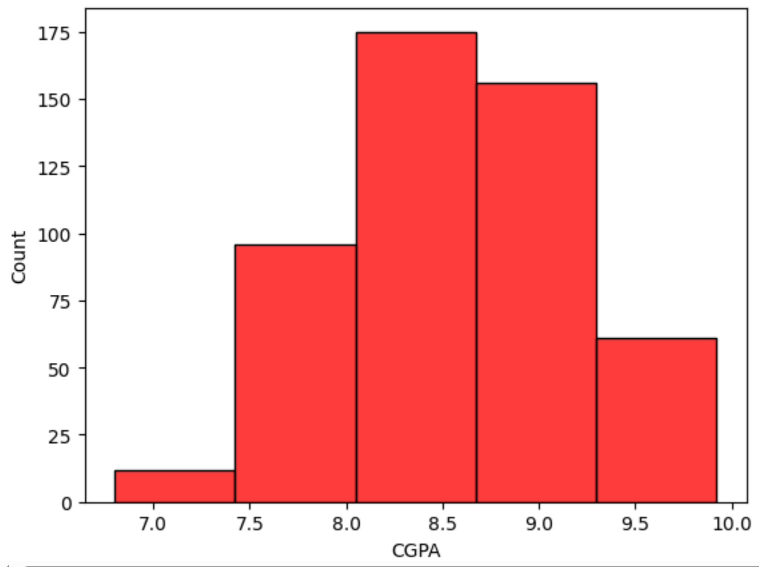
```
sns.countplot(x='SOP',data=df,color='b')
plt.show()
```



```
sns.countplot(x='LOR ',data=df,color='b')
plt.show()
```

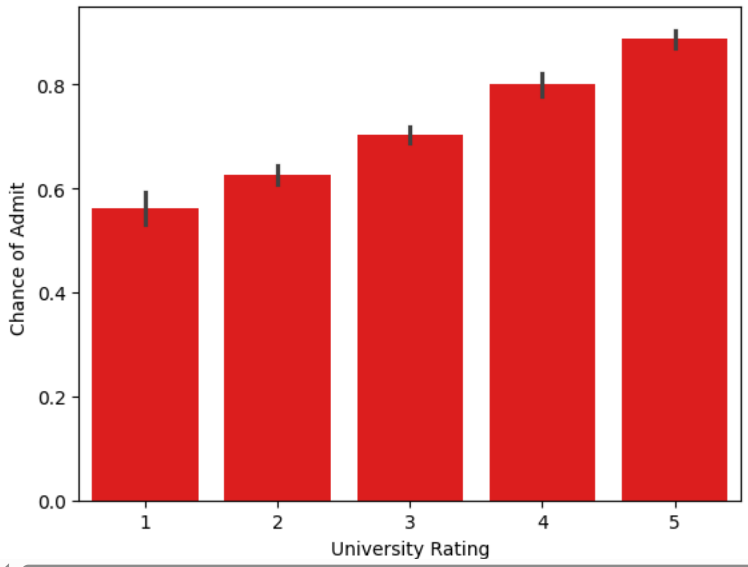


```
sns.histplot(x='CGPA',data=df,color='r',bins=5)
plt.show()
```



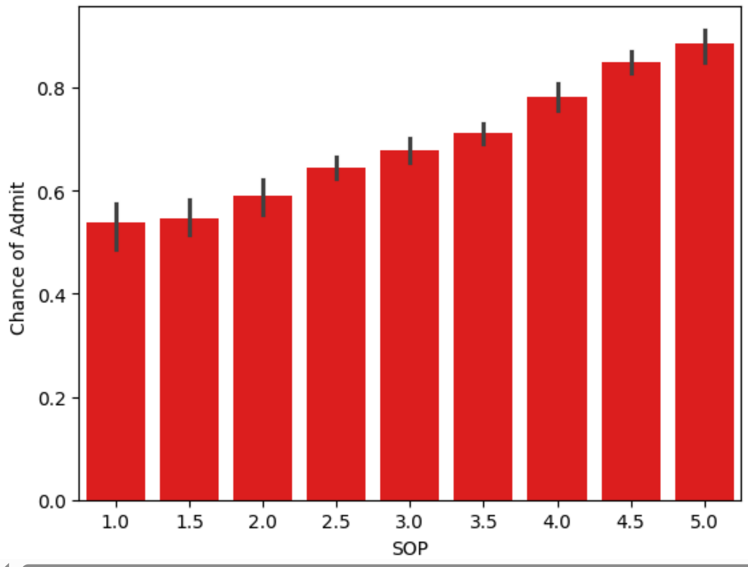
BIVARIATE ANALYSIS

```
sns.barplot(x='University Rating',y='Chance of Admit ',data=df,color='r')
plt.show()
```



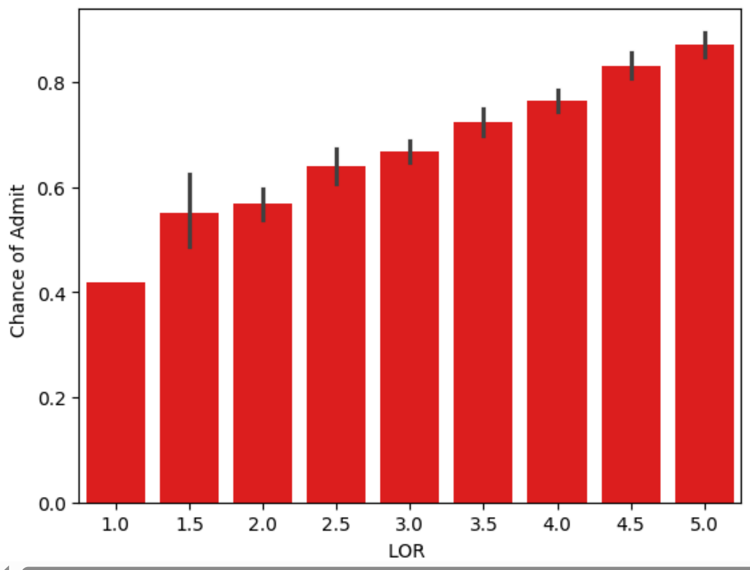
If the **university rating** is **more** the chances of **getting admission** is **more**

```
sns.barplot(x='SOP',y='Chance of Admit ',data=df,color='r')
plt.show()
```



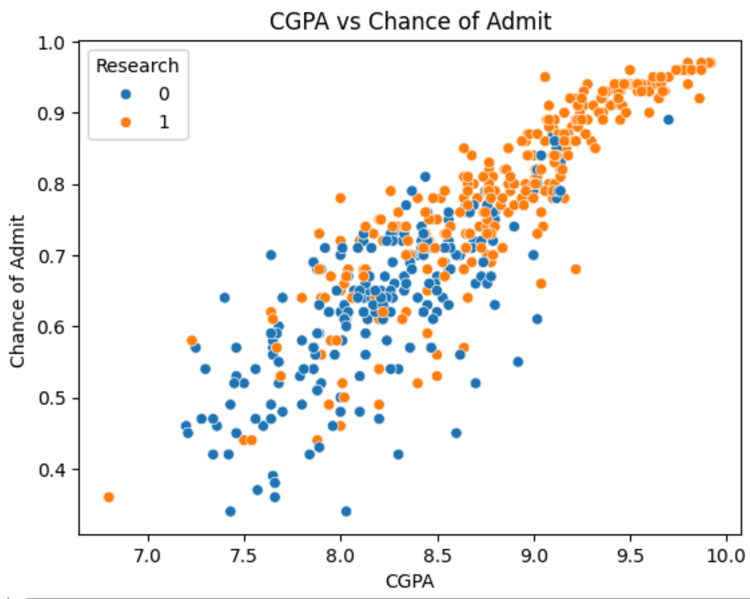
If the **SOP** is **more** the chances of **getting admission** is **more**

```
sns.barplot(x='LOR ',y='Chance of Admit ',data=df,color='r')
plt.show()
```



Same kind of conclusion here also. High LOR results in high chance of getting admission

```
sns.scatterplot(x='CGPA', y='Chance of Admit ', data=df,hue='Research', color='purple')
plt.title('CGPA vs Chance of Admit')
plt.show()
```



From the graph it can be seen clearly that those students who are having high CGPA, their chance of admission is more.One more thing we can observe here is that those students who have high CGPA also have done their Research and also automatically the chance of admission will be more

Now we will do Data Preprocessing

```
#checking for duplicates
print(df.duplicated().sum())
```

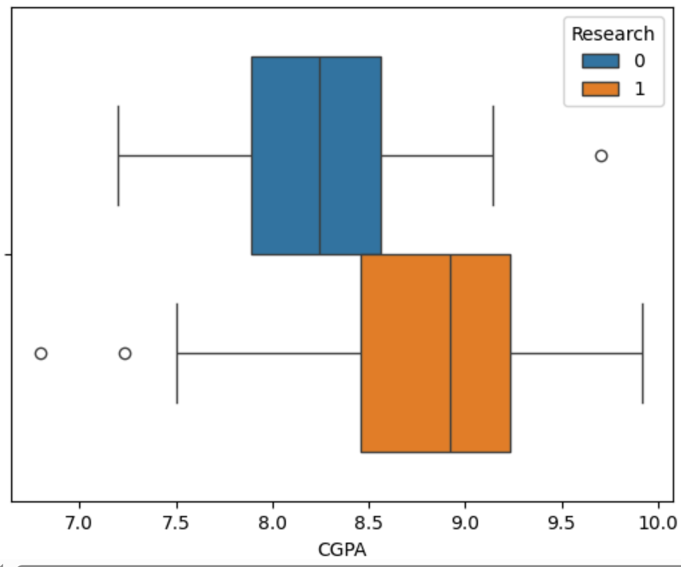
```
0
```

None of the rows are duplicate

```
# as we have seen already that there are no missing value,hence no missing value treatment
```

OUTLIER TREATMENT:

```
sns.boxplot(x='CGPA',hue='Research',data=df)
plt.show()
```



```
from scipy import stats
import numpy as np
z = np.abs(stats.zscore(df[df['Research']==0]['CGPA']))

outlier_indices = np.where(z > 3)[0]        #Using Z-scores for Outlier Detection
df= df.drop(outlier_indices)
print("DataFrame Shape after Removing Outliers:", df.shape)
```

```
DataFrame Shape after Removing Outliers: (499, 9)
```

This part of the code is performing outlier treatment on the dataset. Specifically, it is looking for unusual or extreme values in the 'CGPA' column, but only for the subset of students who have a 'Research' value of 0 (meaning they have not conducted research).

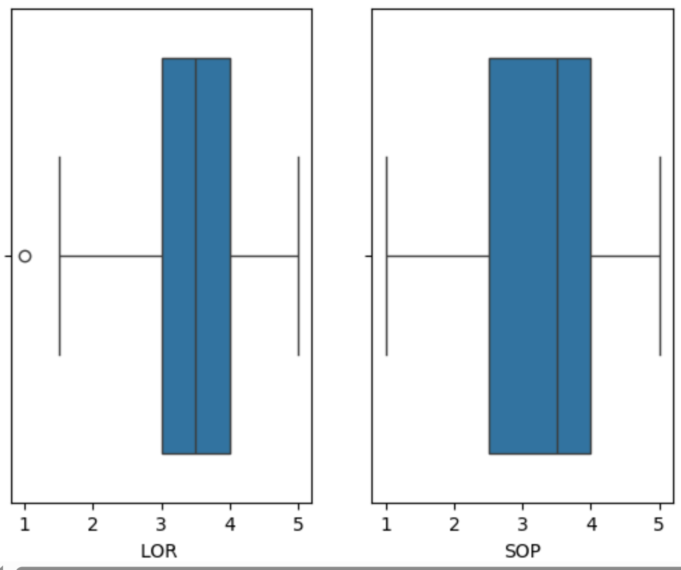
```
z1 = np.abs(stats.zscore(df[df['Research']==1]['CGPA']))

outlier_indices = np.where(z1 > 3)[0]        #Using Z-scores for Outlier Detection
df= df.drop(outlier_indices)
print("DataFrame Shape after Removing Outliers:", df.shape)
```

```
DataFrame Shape after Removing Outliers: (498, 9)
```

This part of the code is performing outlier treatment on the dataset. Specifically, it is looking for unusual or extreme values in the 'CGPA' column, but only for the subset of students who have a 'Research' value of 1 (meaning they have conducted research).

```
plt.subplot(1,2,1)
sns.boxplot(x='LOR ',data=df)
plt.subplot(1,2,2)
sns.boxplot(x='SOP',data=df)
plt.show()
```



we can see there are no outliers for SOP but one outlier for LOR. So we can keep one outlier for data training of model

```
x=df.drop(['Chance of Admit ','Serial No.'],axis=1) #selecting the dependent features as x,
                                          #Serial no. column is of no use,hence removed
y=df['Chance of Admit ']                 #Chance of admit is our target variable,
```

```
#data selection for train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((398, 7), (100, 7), (398,), (100,))
```

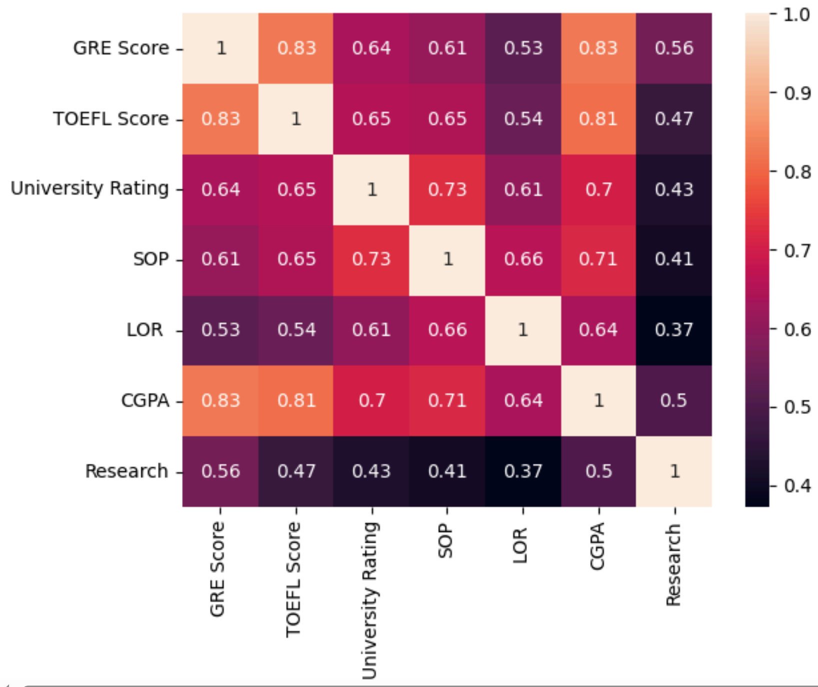
So 398 rows have been choosen for training the model and 100 rows have been choosen to test the model

Standardisation of training data

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train=scaler.fit_transform(x_train)
x_train=pd.DataFrame(x_train,columns=x.columns)
x_test=scaler.transform(x_test)
x_test=pd.DataFrame(x_test,columns=x.columns)
```

```
sns.heatmap(x.corr(),annot=True)
```

```
<Axes: >
```



	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
GRE Score	1	0.83	0.64	0.61	0.53	0.83	0.56
TOEFL Score	0.83	1	0.65	0.65	0.54	0.81	0.47
University Rating	0.64	0.65	1	0.73	0.61	0.7	0.43
SOP	0.61	0.65	0.73	1	0.66	0.71	0.41
LOR	0.53	0.54	0.61	0.66	1	0.64	0.37
CGPA	0.83	0.81	0.7	0.71	0.64	1	0.5
Research	0.56	0.47	0.43	0.41	0.37	0.5	1

Since none of the pair of features have correlation more than 0.90,we are going to keep all the features

Building Linear Regression Model

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(x_train,y_train) # learns the parameters.
```

```
▾ LinearRegression
LinearRegression()
```

Model Coefficiets with column names

```
print('model intercept:',model.intercept_)
data=pd.DataFrame(model.coef_,x.columns,columns=['Coefficient'])
data
```

```
model intercept: 0.7268844221105528
```

	Coefficient
GRE Score	0.025415
TOEFL Score	0.015826
University Rating	0.006915
SOP	0.000952
LOR	0.016667
CGPA	0.067662
Research	0.010921

Ridge and Lasso regression

```
from sklearn.linear_model import Ridge, RidgeCV, Lasso

print("\nRidge Model............................................\n")
#Ridge Regression Model
ridgeReg = Ridge(alpha=.0001)
```

```
ridgeReg.fit(x_train,y_train)

#train and test scorefor ridge regression
train_score_ridge = ridgeReg.score(x_train, y_train)
test_score_ridge = ridgeReg.score(x_test, y_test)
print("The train score for ls model is {}".format(train_score_ridge))
print("The test score for ls model is {}".format(test_score_ridge))
```

```
Ridge Model.........................................

The train score for ls model is 0.8262153957923237
The test score for ls model is 0.7966973412478419
```

```
#Lasso regression model
print("\nLasso Model............................................\n")
lasso = Lasso(alpha = 0.001)
lasso.fit(x_train,y_train)
train_score_ls =lasso.score(x_train,y_train)
test_score_ls =lasso.score(x_test,y_test)

print("The train score for ls model is {}".format(train_score_ls))
print("The test score for ls model is {}".format(test_score_ls))
```

```
Lasso Model.........................................

The train score for ls model is 0.8261323119094164
The test score for ls model is 0.796330354513646
```

```
y_pred = model.predict(x_train)
y_pred[:10]
```

```
array([0.43565201, 0.80482085, 0.74410411, 0.83327356, 0.78313378,
       0.77191785, 0.95163871, 0.4769205 , 0.76629613, 0.68979623])
```

```
y_train[:10]
```

	Chance of Admit
58	0.36
229	0.82
282	0.81
487	0.79
222	0.76
449	0.79
423	0.94
377	0.47
26	0.76
231	0.74

dtype: float64

```
y_test_pred = model.predict(x_test)
y_test_pred[:10]
```

```
array([0.69548383, 0.77886236, 0.60038744, 0.7757769 , 0.53432789,
       0.72400638, 0.62448724, 0.50554554, 0.65301898, 0.72138187])
```

```
y_test[:10]
```

	Chance of Admit
489	0.65
75	0.72
233	0.64
177	0.82
239	0.59
426	0.71
157	0.65
57	0.46
324	0.67
9	0.45

dtype: float64

```
##Multicollinearity check by VIF score
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif = pd.DataFrame()
vif['Features'] = x_train.columns
vif['values'] = [variance_inflation_factor(x_train.values, i) for i in range(x_train.shape[1])]
# round off to 2 decimal
vif['values'] = vif['values'].round(2)
# sort in decreasing order of vif
vif = vif.sort_values(by='values', ascending=False)
vif
```

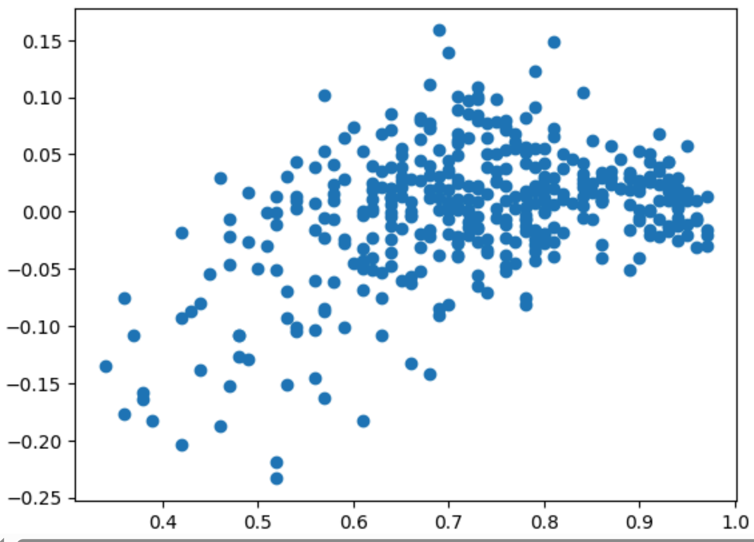
	Features	values
5	CGPA	4.68
0	GRE Score	4.20
1	TOEFL Score	3.70
3	SOP	2.95
2	University Rating	2.66
4	LOR	2.09
6	Research	1.53

As we can see none of the features have VIF more than 5,so we will not remove any features from our training data

Test for Homoscedasticity

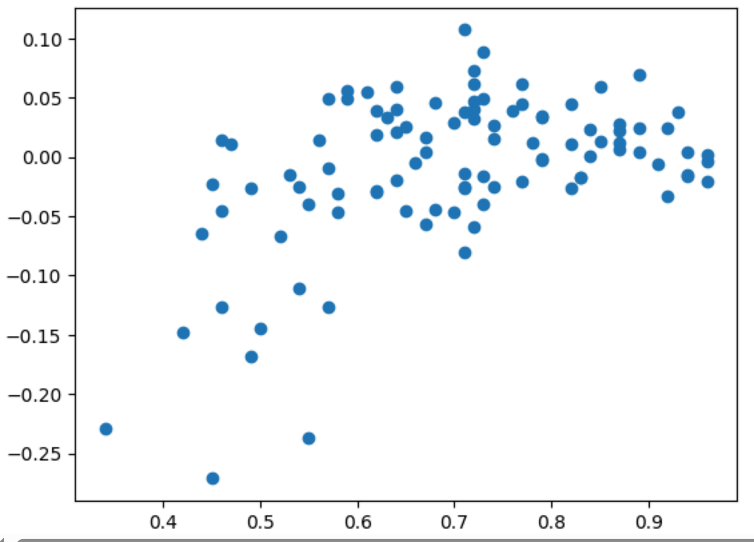
```
errs=y_train-y_pred
plt.scatter(y_train,errs)        #Homoscedasticity for training data
```

```
<matplotlib.collections.PathCollection at 0x7892fe0da9d0>
```

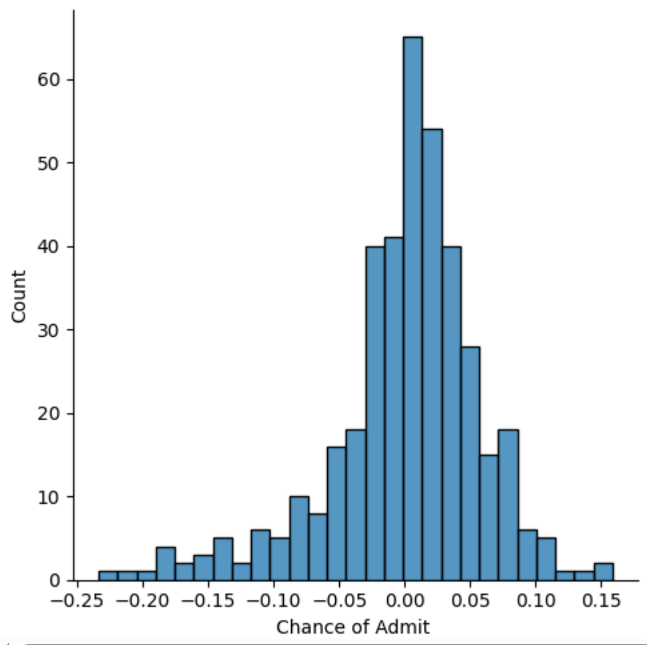


```
errs1=y_test-y_test_pred
plt.scatter(y_test,errs1)                #Homoscedasticity for test data
```

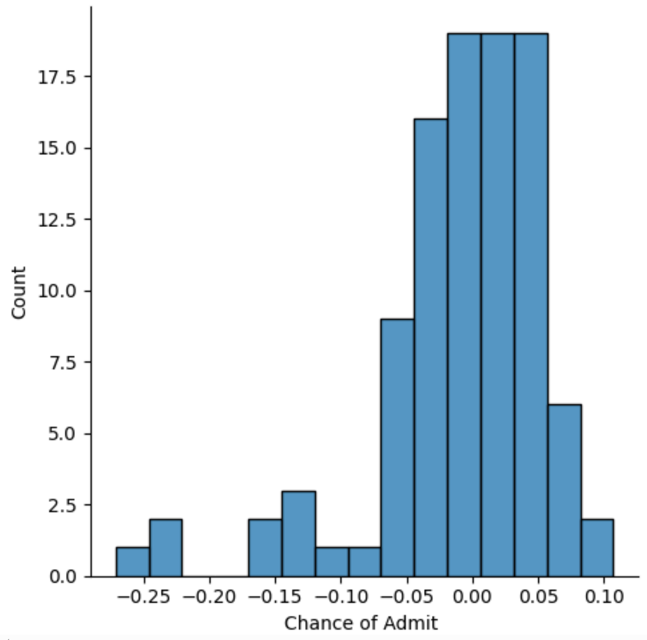
```
<matplotlib.collections.PathCollection at 0x7892f729afd0>
```



```
#NORMALITY OF RESIDUALS FOR THE TRAINING DATA
errs=y_train-y_pred
sns.displot(errs)
plt.show()
```



```
#NORMALITY OF RESIDUALS FOR THE TEST DATA
errs1=y_test-y_test_pred
sns.displot(errs1)
plt.show()
```



Looking at the graph of errors with target, we can say that the errors are normally distributed

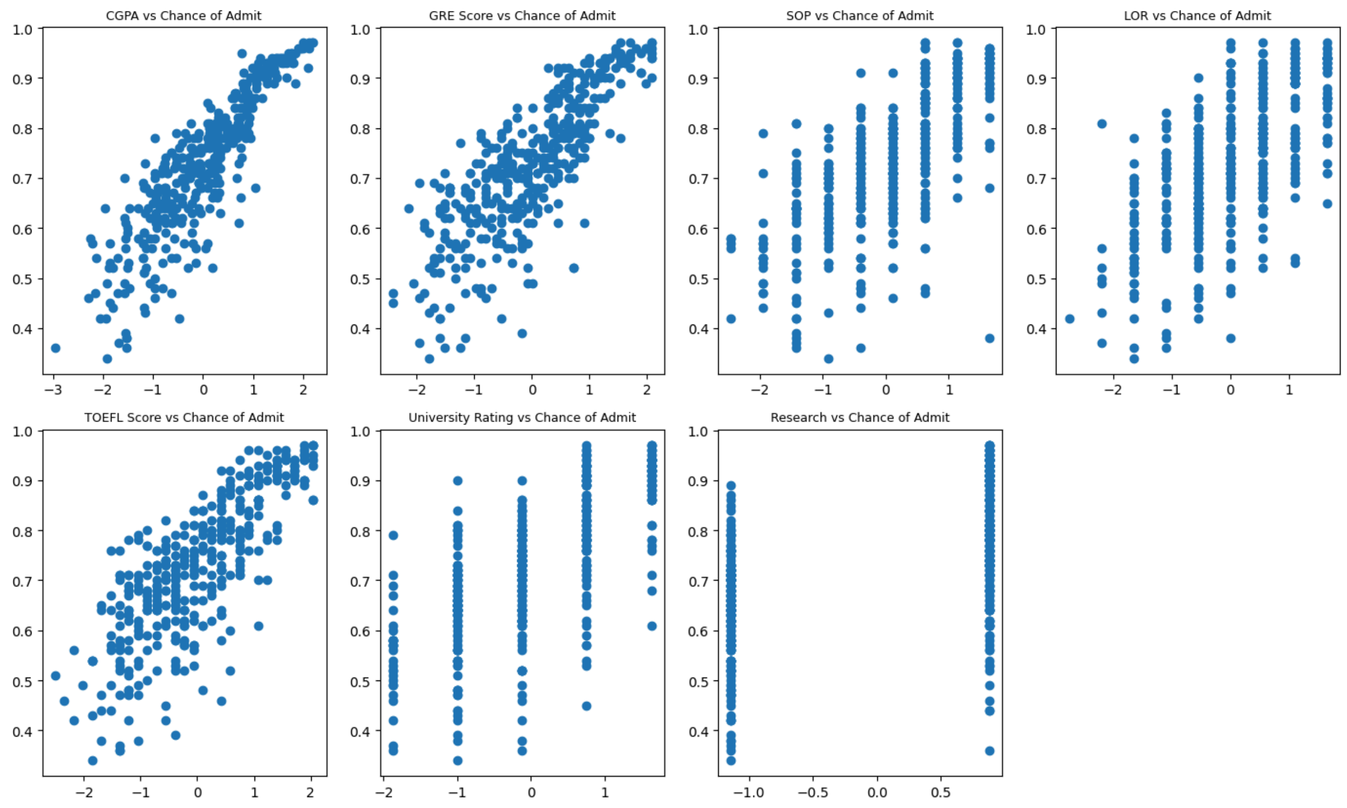
```
a=np.mean(errs1)
print('The mean of errors for test data:',a)
```

```
The mean of errors for test data: -0.00925756314705087
```

we can see the mean of errors is close to zero which indicates that the models are good enough to predict

LINEARITY OF VARIABLES WITH TARGET

```
plt.figure(figsize=(14, 8.4)) # Example: setting a rectangular figure size
plt.subplots_adjust(wspace=0.5, hspace=5)
plt.subplot(2,4,1)
plt.scatter(x_train['CGPA'],y_train)
plt.title('CGPA vs Chance of Admit',fontsize=9)
plt.subplot(2,4,2)
plt.scatter(x_train['GRE Score'],y_train)
plt.title('GRE Score vs Chance of Admit',fontsize=9)
plt.subplot(2,4,3)
plt.scatter(x_train['SOP'],y_train)
plt.title('SOP vs Chance of Admit',fontsize=9)
plt.subplot(2,4,4)
plt.scatter(x_train['LOR '],y_train)
plt.title('LOR vs Chance of Admit',fontsize=9)
plt.subplot(2,4,5)
plt.scatter(x_train['TOEFL Score'],y_train)
plt.title('TOEFL Score vs Chance of Admit',fontsize=9)
plt.subplot(2,4,6)
plt.scatter(x_train['University Rating'],y_train)
plt.title('University Rating vs Chance of Admit',fontsize=9)
plt.subplot(2,4,7)
plt.scatter(x_train['Research'],y_train)
plt.title('Research vs Chance of Admit',fontsize=9)
plt.tight_layout()
plt.show()
```



It can be seen that CGPA,GRE Score,TOEFL Score are having a good linear relationship with target which is chance of admit, but other features are not having that good relationship with target. But in general we can say that for increasing value of features, the target is high

R2 Score,Adjusted R2 Score

```
r2s_test=model.score(x_test,y_test)
adjusted_r2s_test=1-(1-r2s_test)*(x_test.shape[0]-1)/(x_test.shape[0]-x_test.shape[1]-1
print('r2_score for test data:',r2s_test)
```

```
r2_score for test data: 0.7966973584503803
adjusted r2_score for test data: 0.7812286792020396
```

```
r2s_train=model.score(x_train,y_train)
adjusted_r2s_train=1-(1-r2s_train)*(x_train.shape[0]-1)/(x_train.shape[0]-x_train.shape[1]-1)
print('r2_score for train data:',r2s_train)
print('adjusted r2_score for test data:',adjusted_r2s_train)
```

```
r2_score for train data: 0.8262153957923667
[illegible]
```

Comments on the Significance of Predictor Variables

Based on the analysis, several predictor variables show a significant relationship with the 'Chance of Admit'. The linear regression model coefficients also provide insights into the strength and direction of these relationships:

CGPA: The scatter plot clearly shows a strong positive linear relationship between CGPA and the Chance of Admit. The positive coefficient from the linear regression model confirms that higher CGPA scores are associated with a higher chance of admission. This is often the most significant factor in graduate admissions. GRE Score and TOEFL Score: Similar to CGPA, both GRE and TOEFL scores also exhibit a positive linear relationship with the Chance of Admit, although potentially less strong than CGPA. The positive coefficients for these variables indicate that better performance on these standardized tests increases the likelihood of admission. University Rating: The bar plot shows a clear trend: as the university rating increases, so does the average Chance of Admit. This categorical variable is an important indicator of the prestige and quality of the undergraduate institution, which is considered by admissions committees. SOP and LOR: Both Statement of Purpose (SOP) and Letter of Recommendation (LOR) appear to have a positive impact on the Chance of Admit. The bar plots indicate that higher scores in these areas correspond to increased admission probabilities. These factors highlight the importance of a strong personal statement and positive endorsements from faculty. Research: The scatter plot differentiating by 'Research' indicates that students who have conducted research generally have a higher chance of admission, especially at higher CGPA levels. The coefficient for this variable in the linear model quantifies this positive influence. Having research experience demonstrates a commitment to academic inquiry and can be a valuable asset.

The coefficients of the linear regression model provide a numerical representation of the impact of each standardized variable on the 'Chance of Admit', holding other variables constant. Variables with larger absolute coefficient values have a greater influence on the predicted outcome.

Comments on Additional Data Sources for Model Improvement, Model Implementation in the Real World, and Potential Business Benefits

Additional Data Sources for Model Improvement: Qualitative Data on SOP and LOR: While the current dataset includes numerical ratings for SOP and LOR, incorporating qualitative analysis of the content could provide deeper insights. Techniques like Natural Language Processing (NLP) could be used to analyze the themes, tone, and quality of the text in SOPs and LORs. Details about Undergraduate Program/Major: The specific field of study and the rigor of the undergraduate program could significantly influence admission chances for certain graduate programs. Including this information would allow for more nuanced analysis. Extracurricular Activities and Work Experience: Many graduate programs consider a student's involvement in extracurricular activities, internships, or relevant work experience. This information could provide a more holistic view of the applicant. Interview Performance: If applicable, including data on interview performance could further improve the model's predictive power, especially for programs that emphasize this aspect of the application. Applicant Demographics: While sensitive, certain demographic information (e.g., socio-economic background, first-generation student status) could potentially be correlated with factors influencing admissions and might be considered in a comprehensive model (though ethical considerations are paramount here). Information about the Target Graduate Programs: Data on the specific requirements, competitiveness, and yield rates of the target IVY league colleges could be crucial for tailoring predictions. Model Implementation in the Real World: User Interface Integration: The predictive model would need to be integrated into a user-friendly interface on the Jamboree website. This would allow students to input their information and receive an estimated chance of admission. Clear Explanations: The results should be presented clearly, explaining which factors are most positively or negatively impacting their chances. This empowers students to understand where they can improve. Handling Missing or Incomplete Data: The system needs to be robust enough to handle cases where students may not have all the required information. Imputation techniques or providing estimates with less certainty could be considered. Regular Model Updates: Admissions trends and criteria can change over time. The model should be regularly retrained and updated with new data to maintain its accuracy. Ethical Considerations: Ensure the model is used responsibly and does not perpetuate biases. Transparency in how the model works and its limitations is crucial. Integration with Counseling Services: The predictions could be a starting point for discussions with Jamboree counselors, providing personalized guidance based on the model's output and other qualitative factors. Potential Business Benefits from Improving the Model: Enhanced User Engagement: A highly accurate and insightful admission prediction feature would attract more students to the Jamboree website and encourage them to use their services. Increased Conversion Rates: By providing valuable insights and demonstrating their expertise, Jamboree can build trust with potential students, leading to higher conversion rates for their GMAT, GRE, or SAT programs and counseling services. Improved Customer Satisfaction: Accurate predictions and helpful guidance will lead to more satisfied students who feel well-supported in their application journey. Stronger Brand Reputation: A reputation for providing accurate and data-driven insights will position Jamboree as a leader in the overseas education consulting market.

Data-Driven Business Strategy:

The insights gained from the model can inform Jamboree's marketing strategies, course content development, and counseling approaches, allowing them to better cater to the needs of their target audience.
Identification of High-Potential Students: The model could help Jamboree identify students with a high potential for admission,
allowing them to offer targeted support and resources.