University Admit Eligibility Predictor

# >>Import the Libraries

In [1]:

**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**%matplotlib** inline

**import** scipy.stats

**import** statsmodels.api **as** sm

**import** statsmodels.stats.api **as** sms

**import** statsmodels.formula.api **as** smf

**from** statsmodels.stats.stattools **import** jarque\_bera

**from** sklearn.preprocessing **import** OneHotEncoder, StandardScaler

**from** sklearn.decomposition **import** PCA

**from** sklearn.pipeline **import** Pipeline

**from** sklearn.compose **import** ColumnTransformer, make\_column\_selector

**from** sklearn.model\_selection **import** train\_test\_split, cross\_val\_score, GridSearchCV, Ran

**from** sklearn.neighbors **import** KNeighborsRegressor

**from** sklearn.linear\_model **import** Ridge, LogisticRegression

**from** sklearn.svm **import** SVR

**from** sklearn.tree **import** DecisionTreeRegressor

**from** sklearn.ensemble **import** RandomForestRegressor, GradientBoostingRegressor

**from** xgboost **import** XGBRegressor

**from** sklearn.metrics **import** mean\_absolute\_error, mean\_squared\_error, r2\_score

**import** warnings warnings**.**filterwarnings('ignore')

sns**.**set(font\_scale**=**1.5)

# >>Reading the Dataset

In [2]:

*# read\_csv()>>reads the dataset from the csv file.* data **=** pd**.**read\_csv(r'Dataset/Admission\_Predict.csv') data

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[2]: |  | **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 |
|  | **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
|  | **395** | 396 | 324 | 110 | 3 | 3.5 | 3.5 | 9.04 | 1 | 0.82 |
|  | **396** | 397 | 325 | 107 | 3 | 3.0 | 3.5 | 9.11 | 1 | 0.84 |
|  | **397** | 398 | 330 | 116 | 4 | 5.0 | 4.5 | 9.45 | 1 | 0.91 |
|  | **398** | 399 | 312 | 103 | 3 | 3.5 | 4.0 | 8.78 | 0 | 0.67 |
|  | **399** | 400 | 333 | 117 | 4 | 5.0 | 4.0 | 9.66 | 1 | 0.95 |

400 rows × 9 columns

# >>Analyze The Data

In [3]:

*# head() function used to view the first five csv file data.*

data**.**head()

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[3]: |  | **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 |

In [4]:

*# Serial number column is unwanted data for prediction of data. # drop() function delete the Serial No. column from data.*

data**.**drop(["Serial No."],axis **=** 1 ,inplace**=True**) data**.**head()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[4]: |  | **GRE Score** | **TOEFL Score** | **University Rating** | **SOP** | **LOR** | **CGPA** | **Research** | **Chance of Admit** |
|  | **0** | 337 | 118 | 4 | 4.5 | 4.5 | 9.65 | 1 | 0.92 |
|  | **1** | 324 | 107 | 4 | 4.0 | 4.5 | 8.87 | 1 | 0.76 |
|  | **2** | 316 | 104 | 3 | 3.0 | 3.5 | 8.00 | 1 | 0.72 |
|  | **3** | 322 | 110 | 3 | 3.5 | 2.5 | 8.67 | 1 | 0.80 |
|  | **4** | 314 | 103 | 2 | 2.0 | 3.0 | 8.21 | 0 | 0.65 |

In [5]:

*# describe() function computes a summary of statistics like count, mean, standard deviat*

data**.**describe()

Out[5]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Score** | **Rating** |  |  |  |  | **Admit** |
| **count** | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 |
| **mean** | 316.807500 | 107.410000 | 3.087500 | 3.400000 | 3.452500 | 8.598925 | 0.547500 | 0.724350 |
| **std** | 11.473646 | 6.069514 | 1.143728 | 1.006869 | 0.898478 | 0.596317 | 0.498362 | 0.142609 |
| **min** | 290.000000 | 92.000000 | 1.000000 | 1.000000 | 1.000000 | 6.800000 | 0.000000 | 0.340000 |
| **25%** | 308.000000 | 103.000000 | 2.000000 | 2.500000 | 3.000000 | 8.170000 | 0.000000 | 0.640000 |
| **50%** | 317.000000 | 107.000000 | 3.000000 | 3.500000 | 3.500000 | 8.610000 | 1.000000 | 0.730000 |
| **75%** | 325.000000 | 112.000000 | 4.000000 | 4.000000 | 4.000000 | 9.062500 | 1.000000 | 0.830000 |
| **max** | 340.000000 | 120.000000 | 5.000000 | 5.000000 | 5.000000 | 9.920000 | 1.000000 | 0.970000 |

In [6]:

**GRE Score TOEFL**

**University**

**SOP LOR CGPA Research Chance of**

<class 'pandas.core.frame.DataFrame'> RangeIndex: 400 entries, 0 to 399 Data columns (total 8 columns):

*# info() function gives information about the data.*

data**.**info()

# Column Non-Null Count Dtype

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | GRE Score | 400 | non-null |  | int64 |
| 1 |  | TOEFL Score | 400 | non-null |  | int64 |
| 2 |  | University Rating | 400 | non-null |  | int64 |
| 3 |  | SOP | 400 | non-null |  | float64 |
| 4 |  | LOR | 400 | non-null |  | float64 |
| 5 |  | CGPA | 400 | non-null |  | float64 |
| 6 |  | Research | 400 | non-null |  | int64 |
| 7 |  | Chance of Admit | 400 | non-null |  | float64 |

dtypes: float64(4), int64(4) memory usage: 25.1 KB

# >>Handling Missing Values

In [7]:

*# isnull().any() function used to find null values or NaN values in the given dataset by*

data**.**isnull()**.**any()

Out[7]:

In [8]:

GRE Score False

TOEFL Score False University Rating False SOP False

LOR False

CGPA False

Research False

Chance of Admit False dtype: bool

*# isnull().sum() function to check the number of null values present in the columns.*

data**.**isnull()**.**sum()

Out[8]:

In [9]:

cols **=** 3

rows **=** 4

num\_cols **=** data**.**select\_dtypes(exclude**=**'object')**.**columns fig **=** plt**.**figure( figsize**=**(cols**\***5, rows**\***5))

**for** i, col **in** enumerate(num\_cols): ax**=**fig**.**add\_subplot(rows,cols,i**+**1) sns**.**histplot(x **=** data[col], ax **=** ax, data**=**data)

fig**.**tight\_layout() plt**.**show()

GRE Score 0

TOEFL Score 0

University Rating 0

SOP 0

LOR 0

CGPA 0

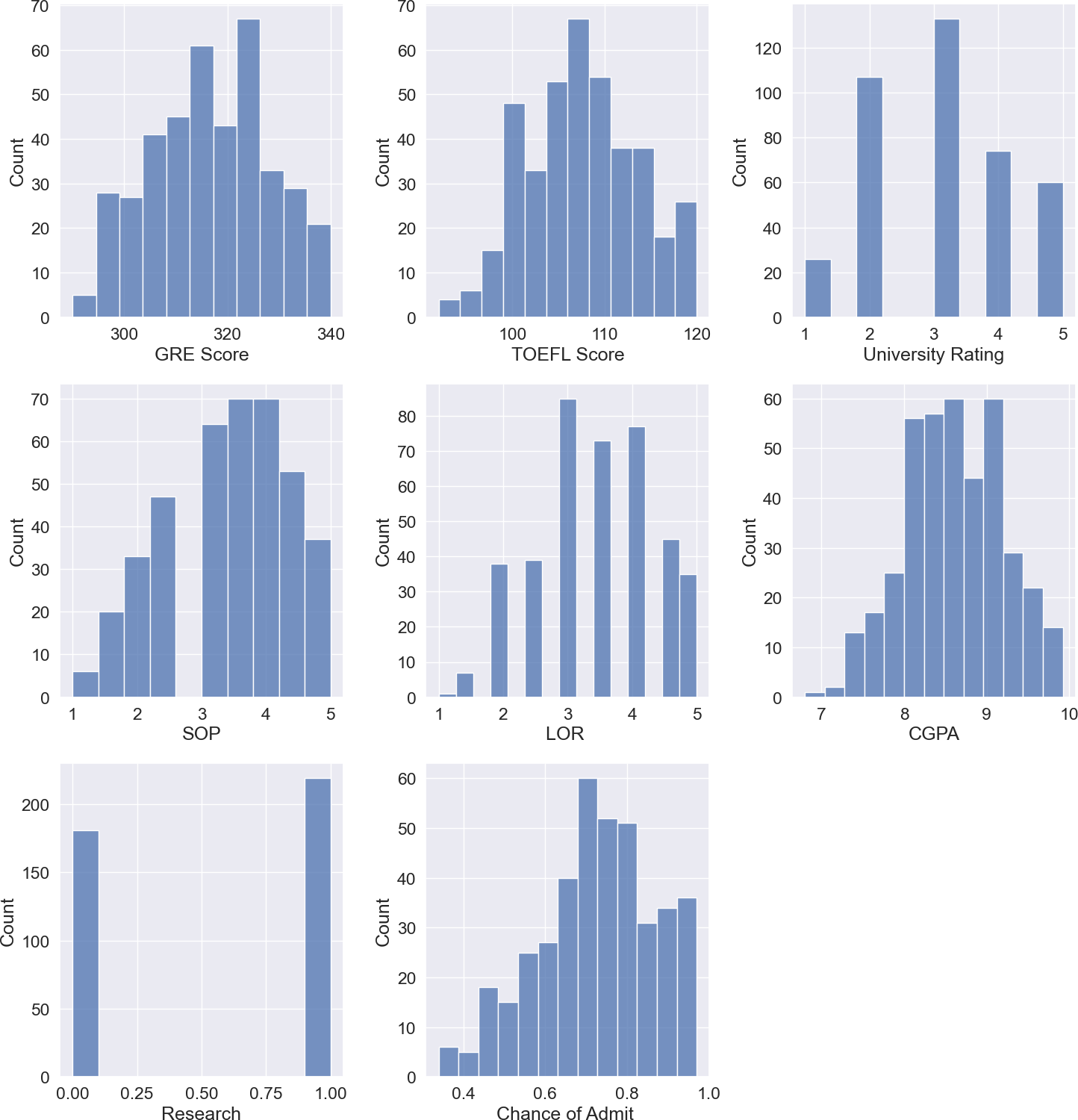
Research 0

Chance of Admit 0

dtype: int64

# >>Data Visualization 1.Univariate Analysis

1. Histogram



# Distplot

In [10]:

cols **=** 3

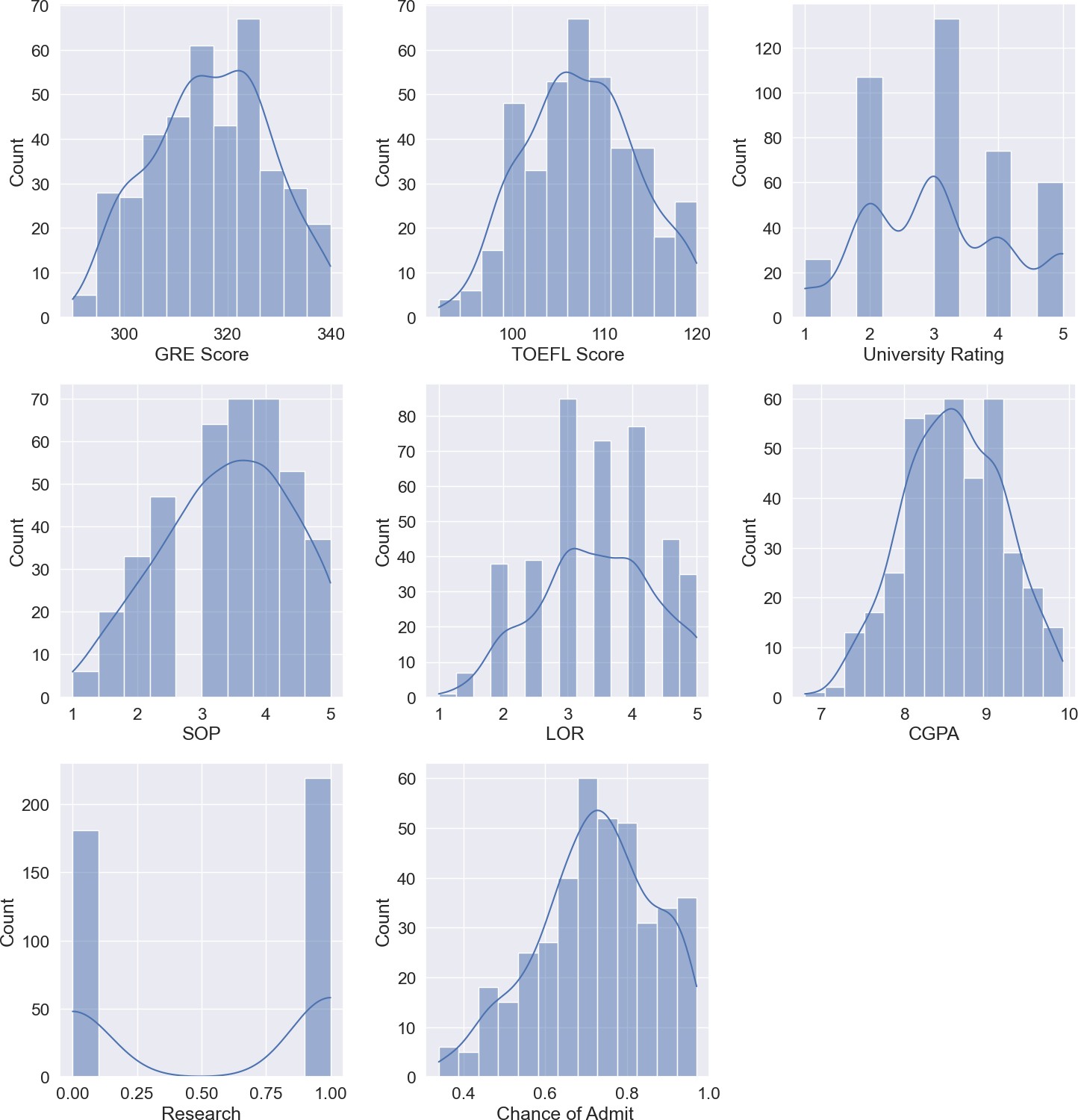
rows **=** 4

num\_cols **=** data**.**select\_dtypes(exclude**=**'object')**.**columns fig **=** plt**.**figure( figsize**=**(cols**\***5, rows**\***5))

**for** i, col **in** enumerate(num\_cols): ax**=**fig**.**add\_subplot(rows,cols,i**+**1)

sns**.**histplot(x **=** data[col], ax **=** ax, data**=**data, kde**=**'True')

fig**.**tight\_layout() plt**.**show()



# Boxplot

In [11]:

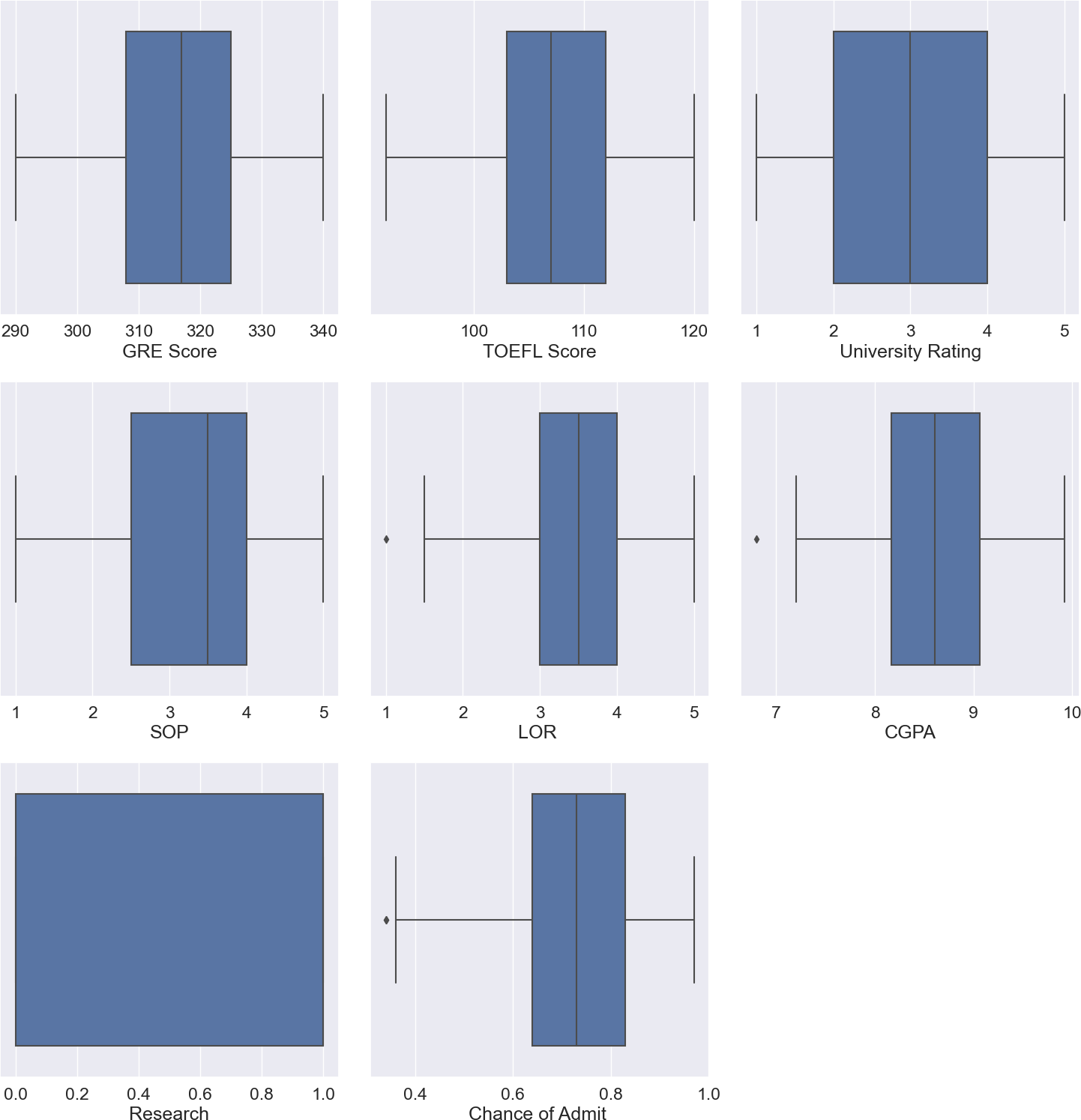
cols **=** 3

rows **=** 4

num\_cols **=** data**.**select\_dtypes(exclude**=**'object')**.**columns fig **=** plt**.**figure( figsize**=**(cols**\***5, rows**\***5))

**for** i, col **in** enumerate(num\_cols): ax**=**fig**.**add\_subplot(rows,cols,i**+**1) sns**.**boxplot(x **=** data[col], ax **=** ax)

fig**.**tight\_layout() plt**.**show()



# 2.Bivariate Analysis

1. Scatterplot

In [12]:

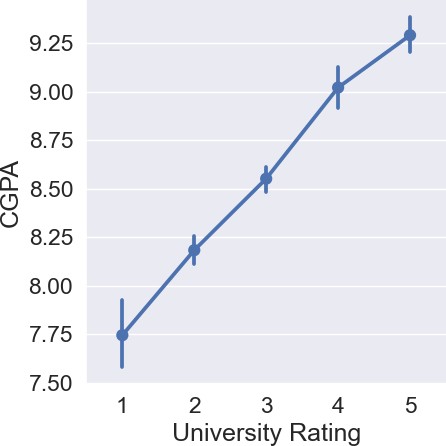
sns**.**scatterplot(x**=**'GRE Score', y**=**'TOEFL Score', data **=** data, color**=**"plum") plt**.**show()



# Catplot

In [13]:

sns**.**catplot(x**=**'University Rating', y**=**'CGPA', data**=**data, kind**=**'point') plt**.**show()



# 2.Multivariate Analysis

I) Pairplot

In [14]:

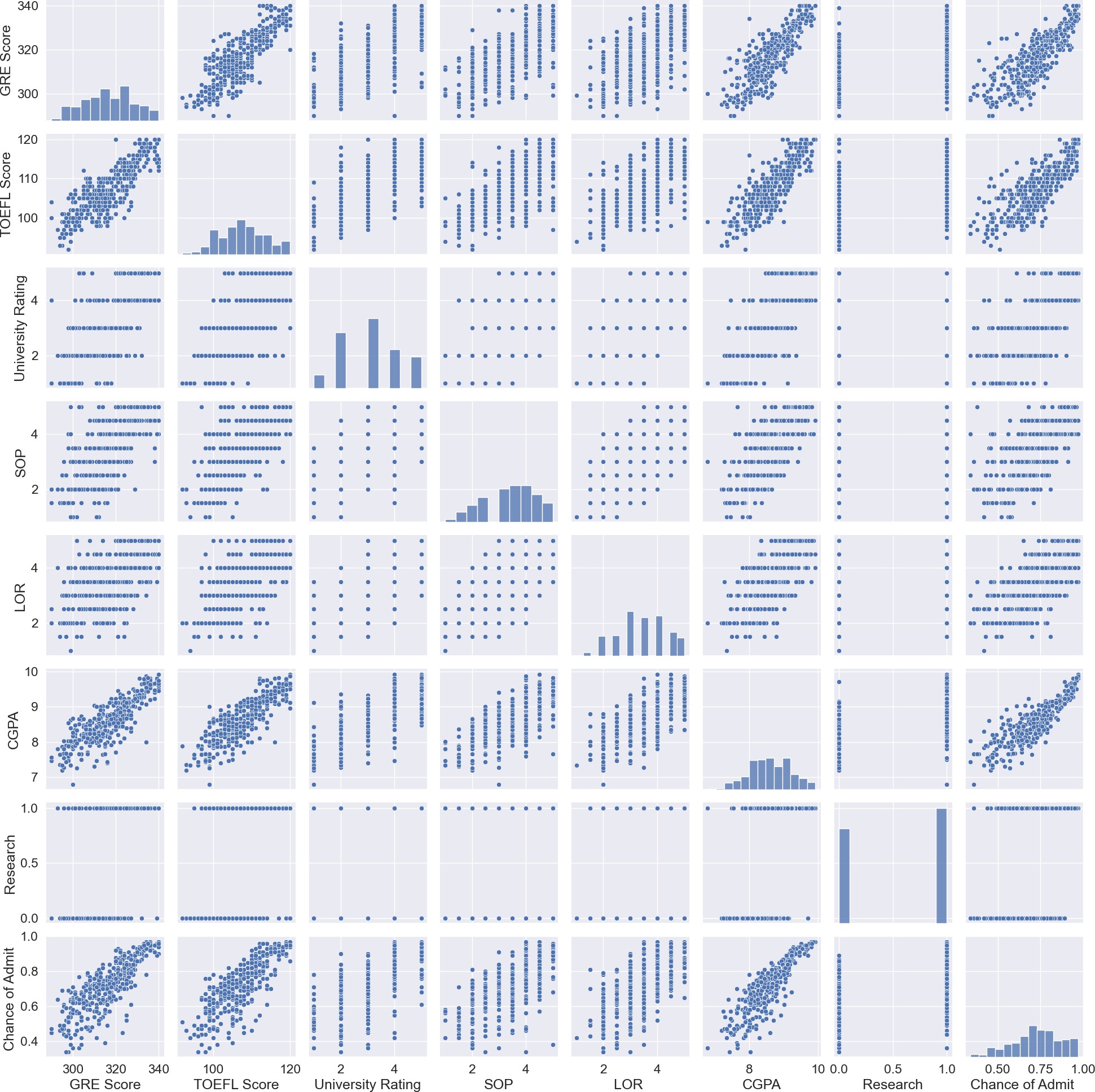
sns**.**pairplot(data**=**data, palette **=** 'Dark2')

Out[14]:

In [15]:

x **=** data**.**iloc[:,0:7]**.**values x

<seaborn.axisgrid.PairGrid at 0x1b38595b430>



# >> Splitting Dependent And Independent Columns

Out[15]:

In [16]:

array([[337.

[324.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| , | 118. | , | 4. | , | ..., | 4.5 | , | 9.65, | 1. | ], |
| , | 107. | , | 4. | , | ..., | 4.5 | , | 8.87, | 1. | ], |
| , | 104. | , | 3. | , | ..., | 3.5 | , | 8. , | 1. | ], |
| , | 116. | , | 4. | , | ..., | 4.5 | , | 9.45, | 1. | ], |
| , | 103. | , | 3. | , | ..., | 4. | , | 8.78, | 0. | ], |
| , | 117. | , | 4. | , | ..., | 4. | , | 9.66, | 1. | ]]) |

[316.

..., [330.

[312.

[333.

y **=** data**.**iloc[:,7:]**.**values y

Out[16]:

array([[0.92],

[0.76],

[0.72],

[0.8 ],

[0.65],

[0.9 ],

[0.75],

[0.68],

[0.5 ],

[0.45],

[0.52],

[0.84],

[0.78],

[0.62],

[0.61],

[0.54],

[0.66],

[0.65],

[0.63],

[0.62],

[0.64],

[0.7 ],

[0.94],

[0.95],

[0.97],

[0.94],

[0.76],

[0.44],

[0.46],

[0.54],

[0.65],

[0.74],

[0.91],

[0.9 ],

[0.94],

[0.88],

[0.64],

[0.58],

[0.52],

[0.48],

[0.46],

[0.49],

[0.53],

[0.87],

[0.91],

[0.88],

[0.86],

[0.89],

[0.82],

[0.78],

[0.76],

[0.56],

[0.78],

[0.72],

[0.7 ],

[0.64],

[0.64],

[0.46],

[0.36],

[0.42],

[0.48],

[0.47],

[0.54],

[0.56],

[0.52],

[0.55],

[0.61],

[0.57],

[0.68],

[0.78],

[0.94],

[0.96],

[0.93],

[0.84],

[0.74],

[0.72],

[0.74],

[0.64],

[0.44],

[0.46],

[0.5 ],

[0.96],

[0.92],

[0.92],

[0.94],

[0.76],

[0.72],

[0.66],

[0.64],

[0.74],

[0.64],

[0.38],

[0.34],

[0.44],

[0.36],

[0.42],

[0.48],

[0.86],

[0.9 ],

[0.79],

[0.71],

[0.64],

[0.62],

[0.57],

[0.74],

[0.69],

[0.87],

[0.91],

[0.93],

[0.68],

[0.61],

[0.69],

[0.62],

[0.72],

[0.59],

[0.66],

[0.56],

[0.45],

[0.47],

[0.71],

[0.94],

[0.94],

[0.57]

Loading [MathJax]/extensions/Safe.js

,

[0.61],

[0.57],

[0.64],

[0.85],

[0.78],

[0.84],

[0.92],

[0.96],

[0.77],

[0.71],

[0.79],

[0.89],

[0.82],

[0.76],

[0.71],

[0.8 ],

[0.78],

[0.84],

[0.9 ],

[0.92],

[0.97],

[0.8 ],

[0.81],

[0.75],

[0.83],

[0.96],

[0.79],

[0.93],

[0.94],

[0.86],

[0.79],

[0.8 ],

[0.77],

[0.7 ],

[0.65],

[0.61],

[0.52],

[0.57],

[0.53],

[0.67],

[0.68],

[0.81],

[0.78],

[0.65],

[0.64],

[0.64],

[0.65],

[0.68],

[0.89],

[0.86],

[0.89],

[0.87],

[0.85],

[0.9 ],

[0.82],

[0.72],

[0.73],

[0.71],

[0.71],

[0.68],

[0.89],

[0.84],

[0.93],

[0.93],

[0.88],

[0.9 ],

[0.87],

[0.86],

[0.94],

[0.77],

[0.78],

[0.73],

[0.73],

[0.7 ],

[0.72],

[0.73],

[0.72],

[0.97],

[0.97],

[0.69],

[0.57],

[0.63],

[0.66],

[0.64],

[0.68],

[0.79],

[0.82],

[0.95],

[0.96],

[0.94],

[0.93],

[0.91],

[0.85],

[0.84],

[0.74],

[0.76],

[0.75],

[0.76],

[0.71],

[0.67],

[0.61],

[0.63],

[0.64],

[0.71],

[0.82],

[0.73],

[0.74],

[0.69],

[0.64],

[0.91],

[0.88],

[0.85],

[0.86],

[0.7 ],

[0.59],

[0.6 ],

[0.65],

[0.7 ],

[0.76],

[0.63],

[0.81],

[0.71],

[0.8 ],

[0.77],

[0.74],

[0.7 ],

[0.71],

[0.93],

[0.85],

[0.79],

[0.76],

[0.78],

[0.77],

[0.9 ],

[0.87],

[0.71],

[0.7 ],

[0.7 ],

[0.75],

[0.71],

[0.72],

[0.73],

[0.83],

[0.77],

[0.72],

[0.54],

[0.49],

[0.52],

[0.58],

[0.78],

[0.89],

[0.7 ],

[0.66],

[0.67],

[0.68],

[0.8 ],

[0.81],

[0.8 ],

[0.94],

[0.93],

[0.92],

[0.89],

[0.82],

[0.79],

[0.58],

[0.56],

[0.56],

[0.64],

[0.61],

[0.68],

[0.76],

[0.86],

[0.9 ],

[0.71],

[0.62],

[0.66],

[0.65],

[0.73],

[0.62],

[0.74],

[0.79],

[0.7 ],

[0.76],

[0.84],

[0.78],

[0.67],

[0.66],

[0.65],

[0.54],

[0.58],

[0.79],

[0.8 ],

[0.75],

[0.73],

[0.72],

[0.62],

[0.67],

[0.81],

[0.63],

[0.69],

[0.8 ],

[0.43],

[0.8 ],

[0.73],

[0.75],

[0.71],

[0.73],

[0.83],

[0.72],

[0.94],

[0.81],

[0.81],

[0.75],

[0.79],

[0.58],

[0.59],

[0.47],

[0.49],

[0.47],

[0.42],

[0.57],

[0.62],

[0.74],

[0.73],

[0.64],

[0.63],

[0.59],

[0.73],

[0.79],

[0.68],

[0.7 ],

[0.81],

[0.85],

[0.93],

[0.91],

[0.69],

[0.77],

[0.86],

[0.74],

[0.57],

[0.51],

[0.89],

[0.95],

[0.79],

[0.39],

[0.38],

[0.34],

[0.47],

[0.56],

[0.71],

[0.78],

[0.73],

[0.82],

[0.62],

[0.96],

[0.96],

[0.46],

[0.53],

[0.49],

[0.76],

[0.64],

[0.71],

[0.84],

[0.77],

[0.89],

[0.82],

[0.84],

[0.91],

[0.67],

[0.95]])

In [17]:

x**.**shape

Out[17]: (400, 7)

In [18]:

y**.**shape

Out[18]:

In [19]:

X\_train,X\_test,Y\_train,Y\_test **=** train\_test\_split(x, y, test\_size **=** 0.2,random\_state **=** 10

(400, 1)

# >>Splitting The Data Into Train And Test

In [20]:

Y\_train **=** (Y\_train**>**0.5) Y\_train

Out[20]:

array([[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[False],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False

Loading [MathJax]/extensions/Safe.js

],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[False],

[False],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True

Loading [MathJax]/extensions/Safe.js

],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[False],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True]])

In [21]:

Y\_test **=** (Y\_test**>**0.5) Y\_test

Out[21]:

array([[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[False],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[ True],

[False]])

# >>Model Building

1. Training And Testing The Model

In [22]:

cls **=** LogisticRegression(random\_state **=** 0)

In [23]:

cls**.**fit(X\_train,Y\_train)

Out[23]:

LogisticRegression(random\_state=0)

egression

LogisticR

▾

In [24]:

y\_pred **=** cls**.**predict(X\_test) y\_pred

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[24]: | array([ True, | True, | True, | True, | False, | True, | True, | True, | True, |
|  | True, | True, | True, | True, | True, | True, | True, | True, | True, |
|  | True, | True, | True, | True, | True, | True, | True, | True, | True, |
|  | True, | True, | True, | True, | True, | True, | True, | True, | True, |
|  | False, | True, | True, | True, | True, | True, | True, | True, | True, |
|  | True, | True, | True, | True, | True, | True, | True, | True, | True, |
|  | True, | True, | True, | True, | True, | True, | True, | True, | True, |
|  | True, | True, | True, | True, | True, | True, | True, | True, | True, |
|  | True, | True, | True, | True, | True, | True, | True, | True]) |  |

# Model Evaluation

In [25]:

**from** sklearn.metrics **import** accuracy\_score,recall\_score,roc\_auc\_score,confusion\_matrix

In [26]:

print("\nAccuracy score:%f" **%**(accuracy\_score(Y\_test,y\_pred)**\***100)) print("Recall score:%f" **%**(recall\_score(Y\_test,y\_pred)**\***100)) print("ROC score : %f\n" **%**(roc\_auc\_score(Y\_test,y\_pred)**\***100)) print(confusion\_matrix(Y\_test,y\_pred))

Accuracy score:91.250000 Recall score:97.333333 ROC score : 48.666667

[[ 0 5]

[ 2 73]]

# Save The Model

In [28]:

**import** pickle pickle**.**dump(cls,open('university.pkl','wb')) model**=**pickle**.**load(open('university.pkl','rb'))

In [ ]: