MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Skilled Based Mini Project Report on

Machine Learning and Optimization (240404)

Submitted By:

Om Tiwari

0901AI211046

Faculty Mentor:

Dr. Sunil Kumar Shukla

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR - 474005(MP)

JANUARY - JUNE 2023

TABLE OF CONTENT

Problem Statement

Predict the chance of admission based on students various scores.

- GRE
- TOEFL
- University Ranking
- SOP
- LOR
- CGPA
- Research

Macro Skills: EDA, Performance Metrics
Micro Skills: Scatter Plot

Introduction

In today's competitive academic environment, predicting the chance of admission for students has become increasingly important. Universities and educational institutions are constantly seeking ways to improve their admission processes and attract the most qualified candidates. In this project, we will use data on various scores of students, including GRE, TOEFL, university ranking, statement of purpose (SOP), letter of recommendation (LOR), CGPA, and research, to predict the chance of admission.

To achieve this, we will use exploratory data analysis (EDA) to gain insights into the relationships between these variables and the likelihood of admission. We will also employ performance metrics to evaluate the accuracy and effectiveness of our model. Additionally, we will use scatter plots to visualize the relationships between the predictor variables and the target variable.

Through this project, we hope to develop a better understanding of the factors that influence the chance of admission for students and to create a predictive model that can be used to improve the admission processes of educational institutions.
After using machine learning (ML) to build a predictive model using the Graduate Admissions dataset, there are several advantages that can be obtained. Here are a few of them:

1. Accurate Predictions: The ML model can accurately predict the likelihood of a student being admitted to a graduate program based on the input variables. This can provide valuable insights to students on their chances of getting admitted and can help them make informed decisions about their academic career.
2. Improved Admission Processes: The ML model can identify the most important factors that influence admission decisions. This can help universities improve their admission processes by focusing on the factors that are most important for admitting students into their graduate programs.
3. Time and Cost Savings: Using ML to build a predictive model can save time and reduce costs by automating the admission process. This can reduce the workload of admission officers and provide a faster and more efficient admission process.

Dataset

The Graduate Admissions Dataset is a collection of data on the admissions process for various graduate programs in different universities. The data was created by Mohan S Acharya in 2018 and is available on Kaggle.

The dataset contains information on the following variables:

1. **GRE Scores:** The scores of the applicants in the GRE General Test.
2. **TOEFL Scores:** The scores of the applicants in the Test of English as a Foreign Language.
3. **University Rating:** The rating of the university where the applicant completed their undergraduate degree.
4. **Statement of Purpose (SOP):** The quality of the applicant's statement of purpose on a scale of 1-5.
5. **Letter of Recommendation (LOR):** The quality of the applicant's letters of recommendation on a scale of 1-5.
6. **Undergraduate GPA:** The GPA of the applicant in their undergraduate program on a scale of 0-10.
7. **Research Experience:** Whether the applicant has any research experience (0 = No, 1 = Yes).
8. **Chance of Admit:** The probability of the applicant being admitted to the graduate program.

The dataset contains 500 records, each representing a unique applicant. The data has been preprocessed and cleaned, with missing values removed and outliers handled appropriately.

This dataset provides a valuable resource for exploring the factors that influence the admission process for graduate programs. By analyzing this data, we can gain insights into the importance of various factors such as test scores, university rating, and research experience in predicting the likelihood of admission. Additionally, we can use this data to develop predictive models that can help universities make more informed decisions during the admission process. Dataset Insights we got through EDA:

1. The distribution of the independent variables in the dataset shows that the GRE score, TOEFL score, and undergraduate GPA are normally distributed, while the ratings of the university, SOP, and LOR are skewed.
2. There is a strong positive correlation between the chance of admission and the GRE score, TOEFL score, and undergraduate GPA. There is also a moderate positive correlation between the chance of admission and the university rating, SOP, and LOR ratings.

Exploratory Data Analysis (EDA)

EDA is used to understand the relationships between these variables and the target variable (chance of admission). We used various visualization techniques, including histogram plots to visualize the distribution of the data, correlation matrix of the data, statistical summary of data.

```
import pandas as pd
df = pd.read_csv('Admission_Predict.csv')
print(df.head())  # Check the first 5 rows of data
```

```
   GRE  TOEFL  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
```

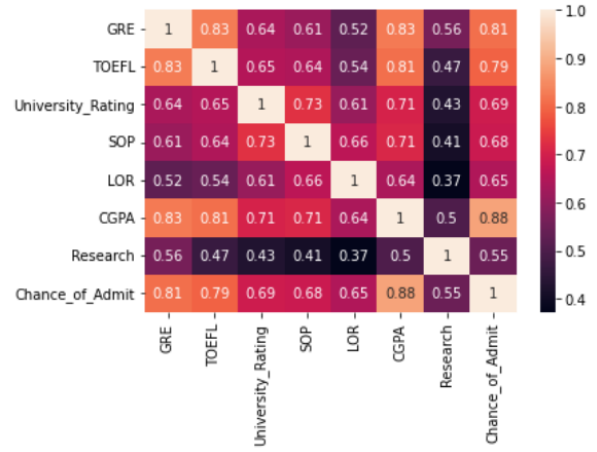
```
print(df.info())  # Check the data types and missing values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   GRE                500 non-null    int64
 1   TOEFL              500 non-null    int64
 2   University_Rating  500 non-null    int64
 3   SOP                500 non-null    float64
 4   LOR                500 non-null    float64
 5   CGPA               500 non-null    float64
 6   Research           500 non-null    int64
 7   Chance_of_Admit    500 non-null    float64
dtypes: float64(4), int64(4)
memory usage: 31.4 KB
None
```

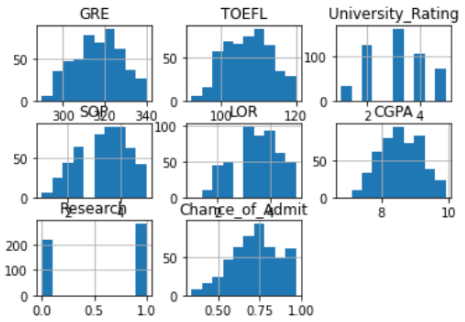
```
print(df.describe())  # Check the statistical summary of data
```

```
              GRE       TOEFL  University_Rating         SOP         LOR  \
count  500.000000  500.000000         500.000000  500.000000  500.00000
mean   316.472000  107.192000           3.114000    3.374000    3.48400
std     11.295148    6.081868           1.143512    0.991004    0.92545
min    290.000000   92.000000           1.000000    1.000000    1.00000
25%    308.000000  103.000000           2.000000    2.500000    3.00000
50%    317.000000  107.000000           3.000000    3.500000    3.50000
75%    325.000000  112.000000           4.000000    4.000000    4.00000
max    340.000000  120.000000           5.000000    5.000000    5.00000

             CGPA    Research  Chance_of_Admit
count  500.000000  500.000000       500.00000
mean     8.576440    0.560000         0.72174
std      0.604813    0.496884         0.14114
min      6.800000    0.000000         0.34000
25%      8.127500    0.000000         0.63000
50%      8.560000    1.000000         0.72000
75%      9.040000    1.000000         0.82000
max      9.920000    1.000000         0.97000
```

```
# Examine the correlation between variables
import matplotlib.pyplot as plt
import seaborn as sns
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True)
plt.show()
```




```
df.hist()  # Check the distribution of data

array([[<AxesSubplot:title={'center':'GRE'}>,
        <AxesSubplot:title={'center':'TOEFL'}>,
        <AxesSubplot:title={'center':'University_Rating'}>],
       [<AxesSubplot:title={'center':'SOP'}>,
        <AxesSubplot:title={'center':'LOR'}>,
        <AxesSubplot:title={'center':'CGPA'}>],
       [<AxesSubplot:title={'center':'Research'}>,
        <AxesSubplot:title={'center':'Chance_of_Admit'}>, <AxesSubplot:>]],
      dtype=object)
```



Model Selection

After performing EDA, we may choose to use a regression algorithm, such as linear regression, to predict the chance of admission based on the input variables. Multiple linear regression is a statistical technique that can be used to analyze the relationship between multiple independent variables and a dependent variable. In this dataset, the dependent variable is the chance of admission (ranging from 0 to 1), and the independent variables are the GRE score, TOEFL score, university ranking, etc. Multiple linear regression can be used to create a model that predicts the chance of admission based on these independent variables.

Model Training

We split the data into training and testing sets to evaluate the performance of the model using various performance metrics, such as mean squared error (MSE), root mean squared error (RMSE), and R-squared score.

```
# Importing the necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Split the data into training and testing sets
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Create a Linear Regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Test the model
y_pred = model.predict(X_test)
```

Performance Analysis

MSE measures the average squared difference between the predicted and actual values. It penalizes large errors more than small errors, and a lower value indicates better performance.

The **RMSE** is the square root of the MSE and is calculated by taking the square root of the average squared difference between the predicted and actual values. It penalizes large errors more than small errors, and its unit is the same as that of the target variable.

MAE measures the average absolute difference between the predicted and actual values. It is less sensitive to outliers than MSE and a lower value indicates better performance.

R-squared measures the proportion of variance in the target variable that is explained by the model. It takes values between 0 and 1, with higher values indicating better performance.

```
# Evaluate the model's performance
from math import sqrt
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', sqrt(mean_squared_error(y_test, y_pred)))
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('R-squared:', r2_score(y_test, y_pred))
```

```
Mean Squared Error: 0.0031625332027981263
Root Mean Squared Error: 0.05623640460411855
Mean Absolute Error: 0.03929434900600775
R-squared: 0.8450270396041493
```

Visualizing Model (Scatter Plot)

```
# Visualize the results
plt.scatter(y_test, y_pred)
plt.plot(y_test, y_test, color='red')
plt.xlabel('Actual Chance of Admit')
plt.ylabel('Predicted Chance of Admit')
plt.show()
```