Assignment No 7

AIM:

Assignment on Classification technique Every year many students give the GRE exam to get admission in foreign Universities. The data set contains GRE Scores (out of 340), TOEFL Scores (out of 120), University Rating (out of 5), Statement of Purpose strength (out of 5), Letter of Recommendation strength (out of 5), Undergraduate GPA (out of 10), Research Experience (0=no, 1=yes), Admitted (0=no, 1=yes). Admitted is the target variable. Data Set: https://www.kaggle.com/mohansacharya/graduate-admissions The counselor of the firm is supposed check whether the student will get an admission or not based on his/her GRE score and Academic Score. So to help the counselor to take appropriate decisions build a machine learning model classifier using Decision tree to predict whether a student will get admission or not. a) Apply Data pre-processing (Label Encoding, Data Transformation) techniques if necessary. b) Perform data-preparation (Train-Test Split) c) Apply Machine Learning Algorithm d) Evaluate Model

THEORY:

Classification

Classification is a supervised learning technique in machine learning where the objective is to predict the categorical class labels of new instances based on past observations. In this case, the classification is binary (0 = Not Admitted, 1 = Admitted).

Decision Tree Algorithm

A Decision Tree is a flowchart-like structure in which each internal node represents a feature (or attribute), each branch represents a decision rule, and each leaf node represents the outcome. Decision Trees can handle both numerical and categorical data and are easy to interpret.

Data Pre-processing
Data pre-processing is a crucial step in machine learning which includes handling missing values, encoding categorical variables, normalizing numerical features, etc. Proper pre-processing ensures better model performance.

Train-Test Split
Dividing the dataset into training and testing parts. The training set is used to build the model, and the test set is used to evaluate its performance.

Evaluation Metrics
- Accuracy: Percentage of correct predictions
- Precision and Recall: Measures of relevance in positive predictions
- Confusion Matrix: Matrix showing actual vs predicted values

METHODOLOGY:

Importing the Dataset
The dataset was downloaded from Kaggle: Graduate Admissions Dataset

Loading and Exploring the Data
We used Python libraries such as Pandas and NumPy to load and explore the data. Key features considered:
- GRE Score
- TOEFL Score
- University Rating
- SOP
- LOR
- CGPA
- Research
- Chance of Admit (converted to Admitted: 0 or 1)

Data Pre-processing
- Label Encoding: Applied on 'Research' column if not binary
- Feature Selection: Chose GRE Score and CGPA for this model
- Target Creation: Created binary target from 'Chance of Admit' using

threshold (e.g., 0.5)
- Data Normalization: Applied MinMaxScaler if necessary

Data Preparation
Splitting the dataset into training and testing sets using an 80:20 ratio with train_test_split from sklearn.

Model Building
Used DecisionTreeClassifier from sklearn.tree. Trained the model on the training set and predicted the target variable on the test set.

Model Evaluation
Used accuracy_score, classification_report, and confusion_matrix to evaluate model performance.

OUTPUTS:

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [2]:
```
df = pd.read_csv("/content/Admission_Predict_Ver1.1.csv")

df.head()
```

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

In [3]:
```
df.drop("Serial No.", axis=1, inplace=True)

df.rename(columns={'Chance of Admit ': 'Chance_of_Admit'}, inplace=True)

df['Admitted'] = df['Chance_of_Admit'].apply(lambda x: 1 if x >= 0.75 else 0)

features = df[['GRE Score', 'CGPA']]
target = df['Admitted']
```

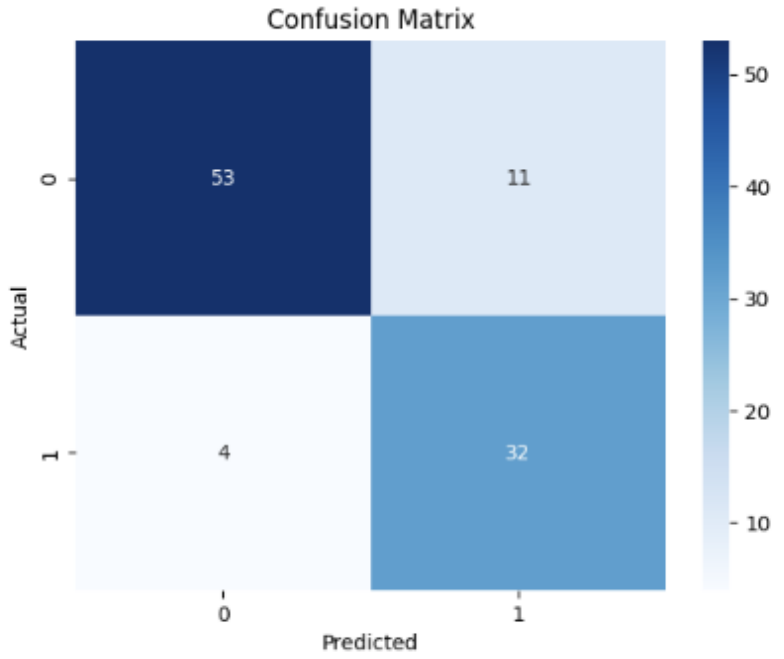
In [4]:
```
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)
```

In [5]:
```
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```

Out[5]:
```
DecisionTreeClassifier(random_state=42)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [6]:
```
y_pred = clf.predict(X_test)

# Accuracy
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy: {acc:.2f}")

# Classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

```
Accuracy: 0.85

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.83      0.88        64
           1       0.74      0.89      0.81        36

    accuracy                           0.85       100
   macro avg       0.84      0.86      0.84       100
weighted avg       0.86      0.85      0.85       100
```



CONCLUSION:

We successfully built a Decision Tree Classifier to predict the admission of students based on their GRE scores and academic performance. From this assignment, we learned the following:

- The importance of data pre-processing and feature selection in improving model performance
- How to implement and train a Decision Tree model
- How to evaluate a classification model using appropriate metrics
- The significance of splitting data into training and testing sets for unbiased evaluation

This assignment provided hands-on experience with classification techniques and model evaluation, which are fundamental concepts in the field of machine learning.