**Name:** Prachi Shende  
 **Roll No:** 281034  
 **Batch:** A2  
 **Assignment 7**

**Statement:**

In this assignment, we apply a Decision Tree Classifier on a dataset related to graduate admissions. The objective is to predict whether a student will get admitted to a foreign university based on GRE and academic scores.

**Objective:**

* Understand and apply the Decision Tree classification algorithm.
* Perform data preprocessing and preparation.
* Evaluate the model using classification metrics.
* Help counselors make informed decisions regarding student admissions.

**Resources Used:**

* **Software used:** Jupyter Notebook
* **Libraries used:** Pandas, NumPy, Matplotlib, Seaborn, Scikit-Learn

**Introduction to Classification:**

Classification is a supervised machine learning technique used to predict categorical labels. In this task, we use the Decision Tree Classifier to predict admission status (admitted: 0 or 1) based on a student’s GRE score and academic profile

**Dataset Description:**

**Source:** [Graduate Admission Dataset on Kaggle](https://www.kaggle.com/mohansacharya/graduate-admissions)

The dataset contains the following features:

* **GRE Score** (out of 340)
* **TOEFL Score** (out of 120)
* **University Rating** (out of 5)
* **SOP** (Statement of Purpose strength - out of 5)
* **LOR** (Letter of Recommendation strength - out of 5)
* **CGPA** (Undergraduate GPA - out of 10)
* **Research** (0 = No, 1 = Yes)
* **Admitted** (0 = No, 1 = Yes) *(Target variable)*

**Methodology:**

1. Data Collection and Exploration:

* Load the dataset into a Pandas DataFrame.
* Explore data for missing values and basic statistics.
* Select relevant features: GRE Score, CGPA, and Admitted.

2. Data Preprocessing:

* Encode the target variable if necessary.
* Scale the selected features for better performance.

3. Data Preparation:

* Perform train-test split (e.g., 80% training, 20% testing).

4. Model Implementation:

* Apply the Decision Tree Classifier.
* Fit the model using training data.
* Predict outcomes on test data.

5. Model Evaluation:

* Use accuracy, confusion matrix, and classification report to evaluate performance.

6. Visualization:

* Visualize the decision tree structure.
* Use heatmaps and bar plots for performance metrics.

**Program Implementation Summary:**

1. **Import Libraries:**
   * pandas, numpy, matplotlib, seaborn, sklearn
2. **Load Data:**
   * Read CSV file and extract GRE Score, CGPA, and Admission Status.
3. **Data Preprocessing:**
   * Check for nulls, drop if any.
   * Feature scaling using StandardScaler.
4. **Train-Test Split:**
   * Split data using train\_test\_split.
5. **Decision Tree Model:**
   * Train using DecisionTreeClassifier.
   * Predict and store results.
6. **Model Evaluation:**
   * Accuracy Score
   * Confusion Matrix
   * Classification Report
7. **Visualization:**
   * Plot the tree structure and evaluation results.

**Results:**

* **Accuracy Score:** *0.87*

**Classification Report:**

precision recall f1-score support

0 0.57 0.40 0.47 10

1 0.92 0.96 0.94 70

accuracy 0.89 80

macro avg 0.74 0.68 0.70 80

weighted avg 0.87 0.89 0.88 80

* **Tree Depth:** *3*
* The decision tree visualization showed clear splits based on GRE and CGPA.

**Advantages:**

* Simple to understand and interpret.
* Requires little data preparation.
* Handles both numerical and categorical data.

**Disadvantages:**

* Prone to overfitting.
* Small changes in data can lead to different tree structures.

**Conclusion:**

This assignment provided hands-on experience with a Decision Tree Classifier. By training the model on GRE and CGPA, we were able to effectively predict the admission status of students. The evaluation metrics demonstrated good performance, and visualizing the tree helped understand the model's decision process. This approach can greatly assist admission counselors in making data-driven decisions.