**Assignment 7**

Name: Yashraj Nimbalkar

Roll No: 281039
Batch: A2

**Problem Statement**

This assignment focuses on applying a Decision Tree Classifier to predict the likelihood of student admission based on features like GRE scores and academic records. The process includes cleaning the dataset, handling categorical data, normalizing values where necessary, splitting the dataset for training and testing, evaluating the model using various classification metrics, and visualizing both the dataset and model outputs through charts and plots.

---

**Objectives**

1. Understand and implement data preprocessing techniques for classification tasks.
2. Build a Decision Tree Classifier using Scikit-Learn and evaluate its effectiveness.
3. Analyze results using standard classification metrics.
4. Use visual tools to explore feature relationships and model output.

---

**Tools and Resources Used**

- **Software:** Visual Studio Code
- **Libraries:** Pandas, Matplotlib, Seaborn, Scikit-Learn

---

**Understanding Classification**

Classification is a form of supervised learning used to assign predefined categories or labels to data points based on input features. Decision Trees work by creating branches based on feature thresholds, resulting in a flowchart-like model that's both simple to understand and effective for many structured datasets, such as student admission records.

---

**Functions and Methods Used**

- pd.read_csv() – Used to read the dataset into a DataFrame.
- .dropna() / .fillna() – Used to manage missing values in the dataset.
- train_test_split() – Splits data into training and test sets.
- DecisionTreeClassifier() – Initializes and trains the decision tree model.
- accuracy_score(), confusion_matrix(), classification_report() – Used to assess the model's performance.

---

**Process Overview**

1. **Data Loading and Initial Analysis**
    - Imported the Graduate Admissions dataset.
    - Reviewed data types, structure, and identified any missing values.
2. **Preprocessing Steps**
    - Filled missing values using mean or median.
    - Converted the target column "Chance of Admit" to binary:
        - 1 for values ≥ 0.5
        - 0 for values < 0.5
    - Selected features like GRE score and CGPA.
    - Normalized values where necessary.
3. **Descriptive Statistics**
    - Used .describe() and NumPy functions to compute key stats:
        - Mean, median, min, max, standard deviation, percentiles.
4. **Visualizations**
    - Created histograms to observe feature distributions.
    - Plotted scatter plots to explore the relationship between features like GRE scores and CGPA.
5. **Model Building and Evaluation**
    - Split the data into 80% training and 20% testing sets.
    - Trained a Decision Tree Classifier using the training data.
    - Made predictions on the test set.
    - Evaluated performance using:
        - Accuracy
        - Confusion Matrix
        - Precision, Recall, and F1-score
    - Visualized the decision tree to interpret the model.

---

**Benefits**

- Pandas simplified data handling and preprocessing.

- Decision Trees provide easy-to-understand results.
- Visualizations revealed patterns and correlations effectively.
- Evaluation metrics offered detailed insights into classification performance.

---

**Drawbacks**

- Can be memory-heavy for large datasets.
- Decision Trees tend to overfit if not properly pruned.
- Performance can degrade with class imbalance in the data.

---

**Conclusion**

In this assignment, we applied a Decision Tree Classifier to predict admission chances based on academic indicators. The workflow included preparing the data, building the model, evaluating results with multiple metrics, and visualizing both the data and model. The use of Python libraries like Pandas, Seaborn, and Scikit-Learn made the entire process efficient and insightful.