Name: Aziz Sayyad
Roll No: 281069
Batch: A3

Practical 7

Problem Statement:

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.

Dataset:

Graduate Admissions Dataset

The dataset consists of the following features:

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

Objectives:

1. Apply **Data Preprocessing** techniques such as Label Encoding and Data Transformation if necessary.
2. Perform **Data Preparation**, including **Train-Test Splitting**.
3. Train a **Decision Tree Classifier** to predict student admissions.
4. Evaluate the model using appropriate classification metrics.

Resources Used:

- **Software:** Visual Studio Code, Jupyter Notebook
- **Libraries:** Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn

Theory:

Classification

Classification is a type of supervised learning where the model assigns labels to data points. In this case, the classifier predicts whether a student will get admitted based on their academic profile. The Decision Tree algorithm is a simple yet powerful classification model that splits the data into subsets based on the most significant features.

Decision Tree Classifier

A Decision Tree is a flowchart-like structure where:

- Internal nodes represent decision points based on feature values.
- Branches represent the outcomes of decisions.
- Leaf nodes represent class labels (admitted or not admitted).

The model splits data recursively to find the best feature that minimizes classification error.

Methodology:

1. Data Preprocessing:

- Load the dataset using Pandas.
- Handle missing values (if any) by imputation or removal.
- Normalize or standardize features for better model performance.
- Convert categorical variables using **Label Encoding** (if required).

2. Train-Test Split:

- Divide the dataset into training (80%) and testing (20%) sets using **train_test_split** from Scikit-learn.

3. Applying Machine Learning Algorithm:

- Use the **DecisionTreeClassifier** from Scikit-learn to train the model on the training data.
- Tune hyperparameters (like max_depth, criterion) for optimal performance.

4. Model Evaluation:

- Generate the **Confusion Matrix** to visualize the model's predictions.
- Compute **Accuracy, Precision, Recall, and F1-score** to assess performance.

Conclusion:

- The **Decision Tree Classifier** successfully predicted student admissions based on their GRE scores and academic performance.
- Performance metrics provided insights into the model's accuracy and reliability.
- Future improvements can include feature engineering and testing other classification models like **Random Forest or Logistic Regression** to enhance accuracy.