Assignment 7

Name: Yashashree Nimbalkar
Roll No: 281040
Batch: A2
PRN: 22311423

Statement

In this assignment, we aim to:

a) Apply Classification Techniques using Decision Trees.
b) Predict whether a student will get admission into a university based on their GRE score, academic performance, and other relevant factors.
c) Implement data preprocessing, including label encoding and data transformation if necessary.
d) Perform data preparation using train-test split.
e) Evaluate the model using appropriate metrics such as accuracy, precision, recall, and F1 score.

Objective

1. Understand the Decision Tree algorithm and its application in classification tasks.
2. Implement data preprocessing techniques like label encoding and normalization.
3. Train and test a classification model on a real-world dataset.
4. Assess the model’s performance using standard evaluation metrics.

Resources Used

- **Software:** VS Code
- **Libraries:** Pandas, NumPy, Matplotlib, Scikit-learn

Introduction to Decision Trees

A Decision Tree is a supervised learning algorithm used for classification and regression tasks. It splits data based on feature values and creates a tree-like model where each node represents a decision rule.

Why Decision Trees?

- Easy to interpret and visualize.
- Can handle both numerical and categorical data.
- Does not require feature scaling or transformation.
- Works well with non-linear relationships.

Dataset Description

We used the Graduate Admission Dataset from Kaggle:

Graduate Admissions Dataset

The dataset contains the following features:

1. **GRE Score** (out of 340)
2. **TOEFL Score** (out of 120)
3. **University Rating** (out of 5)
4. **Statement of Purpose Strength** (out of 5)
5. **Letter of Recommendation Strength** (out of 5)
6. **Undergraduate GPA** (out of 10)
7. **Research Experience** (0 = No, 1 = Yes)
8. **Admitted (Target variable)** (0 = No, 1 = Yes)

Methodology

1. Data Preprocessing

- Checked for missing values and handled them accordingly.
- Applied Label Encoding for categorical variables if necessary.
- Normalized numerical features to ensure a balanced scale.

2. Data Splitting

- Divided the dataset into 80% training data and 20% testing data using train_test_split().

3. Model Training

- Used the **Decision Tree Classifier** from `sklearn.tree`.
- Trained the model using `fit()` on the training data.

4. Model Evaluation

- Calculated classification metrics:
  - **Accuracy Score:** Measures overall correctness.
  - **Precision & Recall:** Evaluates positive class predictions.
  - **F1-Score:** Balances precision and recall.

5. Visualization of Decision Tree

- Used `plot_tree()` from Scikit-learn to visualize the decision-making process.

Advantages of Decision Trees

1. Simple and easy to interpret.
2. No need for feature scaling.
3. Can handle both numerical and categorical data.

Disadvantages

1. Prone to overfitting, especially with deep trees.
2. Sensitive to small variations in data.

Conclusion

In this assignment, we successfully built a Decision Tree classifier to predict student admission outcomes based on GRE scores, academic performance, and research experience. The model was trained and evaluated using classification metrics, and we visualized the decision tree for better understanding. This approach is widely used in education analytics, credit risk assessment, and medical diagnosis.