Assignment 7

Name: Srinidhi Soundarrajan
Roll No: 281064
Batch: A3

Statement

In this assignment, we aim to:
a) Apply a Decision Tree Classifier to predict student admission based on GRE scores and academic performance.
b) Preprocess the dataset by handling missing values, encoding categorical features, and normalizing data if necessary.
c) Split the dataset into training and testing sets for model training and validation.
d) Assess model performance using accuracy, confusion matrix, precision, recall, and F1-score.
e) Visualize the dataset and classification results using histograms, scatter plots, and a decision tree plot.

Objective

1. Understand how to preprocess data for classification tasks.
2. Learn how to build and evaluate a Decision Tree Classifier using scikit-learn.
3. Explore classification metrics to analyze model accuracy.
4. Develop skills in data visualization for better feature interpretation.

Resources Used

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

Introduction to Classification

Classification is a supervised machine learning technique that categorizes data into predefined labels. A Decision Tree Classifier is a tree-based model that splits data into branches based on decision rules, making it interpretable and effective for structured data.

Basic Functions Used

1. pd.read_csv() – Loads data from a CSV file into a DataFrame.
2. .dropna() or .fillna() – Handles missing values using mean, median, or mode imputation.
3. train_test_split() – Splits data into training and testing sets.
4. DecisionTreeClassifier() – Builds a Decision Tree classification model.
5. accuracy_score(), confusion_matrix(), classification_report() – Evaluate classification model performance.

Methodology

1. Data Collection and Exploration

- Load the Graduate Admissions dataset into a Pandas DataFrame.
- Examine its structure, feature types, missing values, and unique categories.

2. Data Preprocessing

- Handle missing values using imputation techniques.
- Convert target variable ("Chance of Admit") to a binary class (1 if admit probability ≥ 0.5, else 0).
- Feature selection: Select relevant features (e.g., GRE Score, CGPA).
- Normalize data if necessary to improve model performance.

3. Summary Statistics Computation

- Compute statistics like minimum, maximum, mean, range, standard deviation, variance, and percentiles using .describe().

4. Feature Visualization using Histograms

- Use histograms (sns.histplot()) to analyze feature distributions.
- Use scatter plots to observe relationships between features (e.g., GRE Score vs. CGPA).

5. Model Building (Decision Tree Classifier)

- Split the dataset into training (80%) and testing (20%) sets using train_test_split().
- Train a Decision Tree Classifier using DecisionTreeClassifier().
- Make predictions on the test set.
- Evaluate the model using accuracy, confusion matrix, precision, recall, and F1-score.

Advantages of Pandas and Classification Analysis

1. Pandas simplifies data handling and provides powerful functions for data preprocessing.
2. Decision Trees are interpretable, making them easy to analyze.
3. Classification metrics help in model evaluation and performance assessment.
4. Data visualization provides insights into feature relationships and class separability.

Disadvantages of Pandas and Data Processing

1. Large datasets can be memory-intensive, slowing down processing.
2. Decision Trees can be prone to overfitting if not pruned or regularized.
3. Imbalanced data may affect classification performance.

Conclusion

This assignment demonstrated the application of a Decision Tree Classifier for predicting student admission based on GRE scores and CGPA. We explored data distributions, computed summary statistics, and evaluated the classification model using various metrics. The use of Pandas, Seaborn, and Scikit-Learn enabled efficient data handling and classification analysis.