Assignment 3

Aim: Implementation of Decision Tree Classifier on Car Evaluation Dataset

Objective:

To implement and evaluate a Decision Tree Classifier using Python to classify car acceptability based on various features.

Introduction:

Importance of Decision Tree:

Decision Trees are a robust supervised learning algorithm for regression and classification tasks. A Decision Tree splits the dataset into branches depending on feature values to make predictions. The objective of this assignment is to preprocess data, train a Decision Tree Classifier, examine its performance, and plot the results on the car evaluation dataset. Decision Trees are a robust supervised learning algorithm for regression and classification tasks. A Decision Tree splits the dataset into branches depending on feature values to make predictions. The objective of this assignment is to preprocess data, train a Decision Tree Classifier, examine its performance, and plot the results on the car evaluation dataset.

Decision Trees are highly prevalent in machine learning due to their simplicity and interpretability. The most significant benefits are:

- **Simple to Interpret and Comprehend:** The tree form is simple to comprehend and interpret.
- **Handles Both Categorical and Numerical Data:** Unlike all other algorithms, Decision Trees can handle mixed data types.
- **Needs Minimal Data Preprocessing:** No feature scaling or heavy data transformation needed.
- **Captures Non-Linear Relationships:** Decision Trees can capture difficult decision boundaries efficiently.
- **Helpful for Feature Selection:** Helpful in choosing meaningful features based on how often they are used to split.

Dataset:

The dataset used in this assignment is **car_evaluation.csv,** which contains categorical attributes influencing car acceptability. The features include:

- **buying**: Buying price of the car
- **maint**: Maintenance cost
- **doors**: Number of doors
- **persons**: Capacity in terms of persons
- **lug_boot**: Size of luggage boot
- **safety**: Safety level
- **class**: Target variable indicating car acceptability (unacc, acc, good, vgood)

Steps of Implementation:

1. **Importing Libraries:**
    - Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, Scikit-Learn, and category_encoders are used for data handling, visualization, and model training.
2. **Loading the Dataset:**
    - Pandas are used to import the data, and a preliminary check is done using.shape(),.head(), and.info() to familiarize ourselves with its structure
3. **Exploratory Data Analysis (EDA):**
    - Show distinct values in categorical features to realize data distribution.
    - Verification for missing values.
4. **Data Preprocessing:**
    - Encoding categorical variables with Ordinal Encoding to convert them into numerical form.
    - Splitting the data set into 67% for training and 33% for testing.
5. **Training the Decision Tree Model:**
    - A Decision Tree Classifier with Gini Index as the splitting criterion and a maximum depth of 3 is used.
6. **Making Predictions:**
    - The trained model makes predictions of car acceptability on the test dataset.
7. **Model Evaluation:**
    - Performance metrics such as Accuracy Score, Confusion Matrix, and Classification Report are calculated.
8. **Visualization of Results:**
    - Plotting the Decision Tree to comprehend decision-making.

Conclusion:

1. Decision Tree Classifier was trained effectively to predict car acceptability.
2. Test set accuracy is 0.8053, which is indicative of good classification performance.
3. The accuracy score of the training set is 0.7848, indicating the model fits well.
4. Confusion Matrix presents information regarding classification mistakes.
5. The Classification Report contains Precision, Recall, and F1-score in order to validate class-wise performance.
6. Decision Tree view depicts decision-making based on input attributes.

References:

https://www.geeksforgeeks.org/decision-tree/

Github:
https://github.com/Pratik-Gadekar123/ML