Assignment 4

Aim: Implementation of Decision Tree Classifier on Car Price Prediction Dataset
Objective: To implement and evaluate a Decision Tree Classifier using Python to predict car price categories based on various features in the CarDekho dataset.

Introduction:

Importance of Decision Tree:
Decision Trees are a powerful supervised learning algorithm for classification and regression tasks. They split the dataset into branches based on feature values to make predictions. In this assignment, we preprocess the **CarDekho dataset**, train a **Decision Tree Classifier**, analyze its performance, and visualize the results.

Advantages of Decision Trees:

- **Easy to Interpret and Understand:** The tree structure is intuitive.
- **Handles Both Categorical and Numerical Data:** Unlike many other algorithms, Decision Trees can process mixed data types.
- **Minimal Data Preprocessing Required:** No need for extensive feature scaling or transformation.
- **Captures Non-Linear Relationships:** Can efficiently model complex decision boundaries.
- **Useful for Feature Selection:** Identifies important variables based on how frequently they are used to split nodes.

Dataset:

The dataset used in this assignment is the **CarDekho dataset**, which contains both numerical and categorical attributes affecting car prices. The key features include:

- **car_name:** Name of the car.
- **brand:** Brand of the car.
- **model:** Model of the car.
- **vehicle_age:** Age of the vehicle.
- **km_driven:** Distance the car has driven in kilometers.
- **seller_type:** Type of the seller (Individual or Dealer).
- **fuel_type:** Type of fuel used (Petrol, Diesel, etc.).
- **transmission_type:** Type of transmission (Manual, Automatic).
- **mileage:** Fuel efficiency of the car.
- **engine:** Engine capacity of the car.
- **max_power:** Maximum power of the car.
- **seats:** Number of seats in the car.
- **selling_price:** Target variable representing the price of the car.

Steps of Implementation:

1. **Importing Libraries:**
    - Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-Learn are used for data handling, visualization, and model training.
2. **Loading the Dataset:**
    - The **CarDekho dataset** is imported using Pandas, and an initial exploration is performed using .shape(), .head(), and .info().
3. **Data Preprocessing:**
    - **Encoding categorical variables** (e.g., brand, fuel_type, transmission_type, seller_type) using **one-hot encoding**.
    - Handling **missing values** by filling categorical columns with **mode** and numerical columns with **median**.
    - **Feature selection** by defining selling_price as the target variable and the other attributes as features.
    - **Splitting the dataset** into 67% training and 33% testing.
4. **Training the Decision Tree Model:**
    - A **Decision Tree Classifier** with **Gini Index** as the splitting criterion and a maximum depth of **3** is used to predict car price categories (e.g., low, medium, high).
5. **Making Predictions:**
    - The trained model is used to **predict car price categories** on the test dataset.
6. **Model Evaluation:**
    - **Accuracy Score:** Measures the overall correctness of the model.
    - **Confusion Matrix:** Provides insights into classification errors, showing false positives and false negatives.
    - **Classification Report:** Displays precision, recall, and F1-score for evaluating model performance.
7. **Visualization of Results:**
    - **Decision Tree Visualization:** The structure of the Decision Tree is plotted to help interpret the decision-making process based on input features.

Conclusion:

- The **Decision Tree Classifier** was effectively trained to predict car prices based on various factors such as mileage, engine capacity, and brand.
- **Accuracy Score**: Indicates how well the model performs in classifying car price categories.
- **Confusion Matrix**: Highlights classification errors (e.g., predicting a car's price category incorrectly).
- **Classification Report**: Provides metrics like precision, recall, and F1-score to assess model performance in detail.

- **Decision Tree Visualization**: Provides an intuitive visual representation of how the model classifies different car price categories based on input features.

References:

- GeeksforGeeks - Decision Tree
- Github Repository: https://github.com/Utkarsh-Rane43/ML-LAB---122B1F110