Car Evaluation Dataset Analysis

Introduction

The car evaluation dataset from the UCI Machine Learning Repository is a well-known dataset used for predicting the quality of cars based on various attributes such as buying price, maintenance cost, number of doors, and safety features. The goal of this analysis is to preprocess the data, train a regression model, and evaluate its performance.

Data Loading and Initial Overview

1. **Libraries Imported**:
   - pandas: For data manipulation and analysis.
   - numpy: For numerical operations.
   - sklearn.model_selection: For splitting the dataset into training and testing sets.
   - sklearn.linear_model: For implementing the Linear Regression model.
   - sklearn.metrics: For calculating the Mean Squared Error.
2. **Dataset Loading**:
   - The dataset is loaded directly from the UCI Machine Learning Repository using pd.read_csv().
   - Column names are assigned based on the dataset documentation.
3. **Missing Values Check**:
   - The initial check indicated no missing values in the dataset.
4. **Dataset Information**:
   - Total Entries: 1728
   - Total Columns: 7
   - Each column is of type object.

5. **Initial Rows Displayed**:
   - The first 5 rows of the dataset show the categorical features related to car evaluation.

Data Cleaning

1. **Column Normalization**:
   - Column names were converted to lowercase and stripped of extra spaces for consistency.
2. **Text Column Cleaning**:
   - All text columns were transformed to lowercase and stripped of whitespace.
3. **One-Hot Encoding**:
   - Categorical columns were converted to a numeric format using one-hot encoding, with the first category dropped to avoid the dummy variable trap.

Target Variable Mapping

1. **Mapping of the evaluation Column**:
   - The target variable evaluation was converted from categorical to numerical format using the following mapping:
     - unacceptable: 0
     - acceptable: 1
     - good: 2
     - very good: 3
2. **NaN Check After Mapping**:
   - A warning was issued indicating the presence of NaN values in the evaluation column post-mapping.
3. **Handling NaNs**:
   - Rows with NaN values in the evaluation column were dropped from the dataset.

Final Dataset Overview

1. **Missing Values After Cleaning**:
   - A final check confirmed no missing values in any column.
2. **Feature and Target Separation**:
   - Features (X) and target variable (y) were separated, with evaluation designated as the target.

Train-Test Split

- The dataset was split into training (80%) and testing (20%) sets using train_test_split().
- The sizes of the training and testing sets were:
  - **Training Set Size**: (55, 15)
  - **Testing Set Size**: (14,15)

Model Training

1. **Model Initialization**:
   - A Linear Regression model was instantiated.
2. **Model Training**:
   - The model was trained on the training data using the fit() method.

Predictions and Evaluation

1. **Predictions**:
   - Predictions were made on the testing set using the trained model.
2. **Model Evaluation**:
   - The model's performance was evaluated using Mean Squared Error (MSE), calculated with mean_squared_error().
   - **Mean Squared Error**: This value indicates the average squared difference between predicted and actual values. (Insert the specific MSE value here).

Conclusion

The analysis of the car evaluation dataset successfully demonstrated the preprocessing and modeling steps necessary for predicting car quality. The dataset was cleaned and transformed into a suitable format for analysis. The Linear Regression model was trained and evaluated, showing its predictive capabilities. While the Mean Squared Error provides an initial measure of model performance, further exploration of more complex models, hyperparameter tuning, and cross-validation is recommended to enhance the predictive accuracy.

References

1. UCI Machine Learning Repository: Car Evaluation Data Set

https://archive.ics.uci.edu/ml/datasets/car+evaluation

2. Pandas Documentation

https://pandas.pydata.org/pandas-docs/stable/

3. Scikit-learn Documentation

https://scikit-learn.org/stable/

4. NumPy Documentation

https://numpy.org/doc/stable/