Data:

Dataset Name: 90,000+ Cars Data From 1970 to 2024

file Name: CarsData2.csv

size: (5.71 MB)

97712 rows

10 columns

link: https://www.kaggle.com/datasets/meruvulikith/90000-cars-data-from-1970-to-2024

intro:

Welcome Here you'll find a collection of over 90,000 used cars spanning from the year 1970 to 2024. This dataset offers a comprehensive glimpse into the world of automobiles, providing valuable insights for researchers, enthusiasts, and industry professionals alike.

Features Included:

(object) Model: The model of the car. (196 unique value)

(int64) Year: The manufacturing year of the car. (1970 TO 2024)

(int64) Price: The price of the car. (159999 to 450)

(object) Transmission: The type of transmission used in the car. (Manual, Semi-Auto, Automatic, Other)

(int64) Mileage: The mileage of the car. (323000 to 1)

(object) FuelType: The type of fuel used by the car. (Petrol, Diesel, Hybrid, Other, Electric)

(int64) Tax: The tax rate applicable to the car. (580 to 0)

(float64)MPG: The miles per gallon efficiency of the car. (470.8 to 0.3)

(float64)EngineSize: The size of the car's engine. (6.6 to 0.0)

(object) Manufacturer: The manufacturer of the car. (ford, volkswagen, vauxhall, merc, BMW, Audi, toyota, skoda, hyundi)

dtypes: float64(2), int64(4), object(4)

Origin:

The dataset is collected from 100,000 UK Used Car Data Set by Aditya :-

Source: Scraped used car listings

Original Collection: The original dataset comprising 100,000 scraped used car listings was collected and cleaned by Aditya

Modification: The dataset was further modified and cleaned to ensure accuracy, consistency, and usability, resulting in the creation of a brand new dataset.

libraries:

1- pandas

2- seaborn

3- matplotlib.pyplot

4- sklearn.pipeline

5- sklearn.preprocessing

6- sklearn.compose

7- joblib

8- sklearn.model\_selection

9- sklearn.tree

10- random

11- scipy.stats

12- sklearn.metrics

Data Preprocessing:

The dataset undergoes preprocessing to prepare it for model training. This involves two main steps:

1- Data Normalization: Numerical features are standardized using StandardScaler() from scikit-learn's preprocessing module. This step ensures that all numerical features have a mean of 0 and a standard deviation of 1.

2- Categorical Encoding: Categorical features are encoded using one-hot encoding (OneHotEncoder()). This transforms categorical variables into binary vectors, making them suitable for machine learning algorithms that require numerical input.

The preprocessing steps are combined into a ColumnTransformer() named preprocess\_pipeline, which applies the appropriate transformations to numerical and categorical features separately.

Data Splitting:

1- Splitted into X and Y:

X: all data - price Column

Y: price column

2- the dataset is split into training and testing sets using train\_test\_split() from scikit-learn's model\_selection module. The split ratio is set to 80% training ()and 20% testing data.

X\_train: (78169, 219)

y\_train: (78169)

X\_test: (19543, 219)

y\_test: (19543)

Model:

Model Training and Hyperparameter Tuning:

Model Training:

A Decision Tree Regressor model is trained using the training data. The hyperparameters for this model are:

1- max\_depth=19

2- min\_samples\_leaf=2

3- min\_samples\_split=13

the R-squared (R2) score is calculated to assess the model's performance on the training data.

The R-squared score on the training data is found to be approximately 0.9691, indicating that the model explains approximately 96.91% Train

Hyperparameter Tuning:

To further optimize the model's performance, hyperparameters are tuned using Randomized Search Cross Validation (RandomizedSearchCV). The hyperparameter space includes:

1- max\_depth(1, 20)

2- min\_samples\_split(2, 20)

3- min\_samples\_leaf(1, 20)

Hyperparameters For RandomizedSearch:

1- tree\_reg

2- param\_distributions=param\_dist

3- n\_iter=100

4- cv=5

5- scoring='r2'

6- random\_state=42

Randomized Search Cross Validation searches for the best combination of hyperparameters by fitting the model multiple times on different subsets of the training data and selecting the combination that maximizes the specified scoring metric, in this case, R-squared (scoring='r2').

After hyperparameter tuning, the best parameters found are :

1- max\_depth=19

2- min\_samples\_leaf=2

3- min\_samples\_split=13

The R-squared score on the test data is found to be approximately 0.9396, indicating that the model explains approximately 93.96% Test