Assignment 2

Aim: Implementation of Linear Regression on CarDekho Dataset

Objective: To implement and evaluate a Linear Regression model using Python to predict car prices based on different features in the dataset.

Theory: Linear Regression is a fundamental supervised learning algorithm used to predict numerical values based on input features. It assumes a linear relationship between the target variable (dependent variable) and the feature variables (independent variables). The objective of this assignment is to perform data preprocessing, train a Linear Regression model, validate its performance, and analyze the results based on the CarDekho dataset.

Importance of Linear Regression:

- **Predictive Analysis:** Helps estimate car prices based on various car attributes.
- **Feature Relationships:** Identifies relationships between vehicle age, mileage, engine power, and car price.
- **Efficiency:** Computationally less expensive and easy to implement.
- **Baseline Model:** Serves as a foundation for more complex regression models.

Dataset: The dataset used for this assignment is **CarDekho Dataset**. It contains various features that describe used cars, such as:

- **Vehicle Age:** Number of years since the car was manufactured.
- **Km Driven:** Total kilometers the car has been driven.
- **Fuel Type:** Type of fuel used (Petrol, Diesel, CNG, Electric, etc.).
- **Transmission Type:** Whether the car has an automatic or manual transmission.
- **Mileage:** The fuel efficiency of the car.
- **Engine:** The engine capacity in CC.
- **Max Power:** The maximum power output of the car.
- **Seats:** The number of seats in the car.
- **Selling Price:** Target variable representing the price at which the car is being sold.

Steps of Implementation:

1. Importing Libraries:

- **Use essential Python libraries for data processing, visualization, and modeling:**
    - **Pandas:** Data handling.
    - **NumPy**: Numerical operations.
    - **Matplotlib & Seaborn**: Visualizations.
    - **Scikit-Learn (sklearn):** Model training and evaluation.

. Loading the Dataset:

- **Load the dataset using pandas.read_csv().**
- **Use .head(), .info(), and .describe() to explore:**
    - Key columns such as vehicle_age, km_driven, fuel_type, transmission_type, mileage, engine, max_power, and selling_price.

3. Data Preprocessing:

- **Handling Missing Values:**
  - Replace missing values in categorical columns (e.g., fuel_type, transmission_type) using the mode.
  - Fill missing values in numerical columns (e.g., mileage, engine, max_power, seats) using the median.
- **Encoding Categorical Variables:**
  - Apply One-Hot Encoding on columns like brand, model, fuel_type, seller_type, and transmission_type.
- **Feature and Target Definition:**
  - Target Variable (y): selling_price
  - Feature Set (X): Exclude car_name, Unnamed: 0, and selling_price.
- **Train-Test Split:**
  - Split the data using train_test_split() from Scikit-Learn (80% Training, 20% Testing).

4. Training the Model:

- Train a Linear Regression model using Scikit-Learn on the training dataset.

5. Making Predictions:

- Use the trained model to predict selling prices on the test dataset.

6. Model Evaluation:

- **Evaluate model performance using metrics from sklearn.metrics:**
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - R^2 Score

7. Visualization of Results:

- Plot a scatter plot using Matplotlib or Seaborn:
  - Compare Actual vs Predicted values of selling_price.

Conclusion:

- **Linear Regression Model Performance:** The model effectively predicts car prices based on key vehicle attributes.
- **Evaluation Metrics:**
  - **MAE:** Measures the average absolute difference between actual and predicted car

prices.

  - **MSE & RMSE:** Indicate the spread of error.
  - **R^2 Score:** Determines how well independent variables explain price variation.

- **Visual Representation:** A scatter plot helps understand the correlation between predicted and actual car prices.