1. Fixing Missing Values

What I did

- I reviewed the dataset to identify which columns had missing entries.
- Numerical fields were filled using the **median**, since it is stable even when extreme values exist.
- Categorical fields were completed using the **mode**, which represents the most frequent category.
- After filling these values, I rechecked the dataset to confirm that no missing cells remained.

Interpretation

- Handling missing values is important because many algorithms cannot process incomplete data.
- Using the median avoids distortion from unusually large or small numbers.
- Replacing empty categories with the most common value helps maintain consistency within each field.
- As a result, the dataset becomes complete and ready for clean analysis or modeling.

2. Removing Units From Columns

What I did

- I built a cleaning function that extracts only numeric characters from strings containing values and units.
- This function was applied to columns such as **Mileage**, **Engine**, **Power**, and **New Price**.
- Each cleaned column was then converted into a numeric type.
- I verified the updated columns to ensure unit labels had been removed correctly.

Interpretation

- Removing units ensures that these fields can be used in calculations without errors.
- Mixed text and numbers can cause type conversion issues and incorrect calculations.
- Converting everything into numeric form makes the dataset more uniform.
- This improves model performance because models work best with clean numerical inputs.

3. Encoding Categorical Variables

What I did

- I identified the columns that contain text labels (categorical variables).
- These columns were transformed into numerical dummy variables using one-hot encoding.
- The encoding process used `drop_first=True` to prevent duplicating information.
- The original text-based columns were replaced with the new encoded variables.

Interpretation

- Many machine-learning models require numerical values instead of text.
- One-hot encoding allows the model to distinguish between different categories.
- Dropping the first dummy column removes redundancy and avoids multicollinearity.
- Overall, this transformation makes the dataset suitable for predictive modeling.

4. Creating a New Feature: Car Age

What I did

- I obtained the current year using Python's date utilities.
- I confirmed that the dataset contained a **Year** column representing when the car was manufactured.
- I calculated the **Car Age** by subtracting the manufacturing year from the current year.
- This newly computed age value was added to the dataset as a separate feature.

Interpretation

- Car age provides clearer insight into depreciation than the raw year of manufacture.
- Older vehicles usually have lower resale prices, so this information is valuable for modeling.
- The new feature helps improve prediction accuracy, especially for price analyses.
- Adding derived features increases the overall usefulness of the dataset.

5. Selecting, Filtering, Renaming, and Summarizing

What I did

- I chose a subset of columns to focus on specific aspects of the data.
- Rows were filtered to highlight records with values above the median of a selected column.
- I renamed one column for clarity and created an additional variable based on simple transformation.
- I sorted the dataset and created grouped summaries using one of the feature columns.

Interpretation

- These steps help isolate meaningful parts of the dataset for deeper inspection.
- Filtering exposes differences between higher-value and lower-value records.
- Renaming columns and creating new features make the dataset easier to read and interpret.
- Group summaries reveal overall trends and support data-driven insights.