**Detailed Preprocessing Documentation for Your Project**

**Dataset Used:**

* Source: segmented\_structured\_dataset.csv
* Initial Rows: **8,300**

**Features Available:**

| **Feature** | **Data Type** | **Description** |
| --- | --- | --- |
| make | Categorical | Specific model name (Alto, Creta, etc.) |
| modelyear | Numerical (Year) | Manufacturing year of the car |
| transmission | Categorical | Manual / Automatic |
| Body Type | Categorical | Hatchback / Sedan / SUV etc. |
| Fuel Type | Categorical | Petrol / Diesel / CNG etc. |
| km | Numerical (Integer) | Kilometers driven by the car |
| ownerno | Numerical (Integer) | Number of previous owners |
| prize | Numerical (Float) | Used car price (target variable, in Lakhs) |
| city | Categorical | City of listing |

**Preprocessing Steps Applied:**

| **Step** | **Feature(s)** | **Method Used** | **Reason** |
| --- | --- | --- | --- |
| Handling Missing Values | modelyear, km, ownerno | Median Imputation | Robust against outliers |
| Handling Missing Values | oem, model, Variant, transmission, Body Type, Fuel Type, city, signature | Mode Imputation (Most Frequent) | Preserve category distributions |
| Handling Missing Values | prize | Median Imputation | No missing target values allowed |
| Encoding Categorical Features | make, transmission, Body Type, Fuel Type, city, signature | One-Hot Encoding (get\_dummies) | To convert categorical to numerical for ML models |
| Scaling Numerical Features | modelyear, km, ownerno, prize | Min-Max Scaling (0 to 1) | To balance different ranges and make training stable |
| Outlier Removal (IQR Method) | km | IQR Outlier Removal | Remove unrealistically high/low driven cars |
| Outlier Removal (IQR Method) | prize | IQR Outlier Removal | Remove wrong price listings |

**Effects of Preprocessing:**

| **Step** | **Effect** |
| --- | --- |
| Handling Missing Values | No NaN or empty entries left |
| Encoding Categorical Features | Model now understands brand, model, variant information numerically |
| Scaling Numerical Features | Features balanced on the same scale without losing real-world meaning |
| Outlier Removal | Reduced dataset to **7,400** rows, removed scrap/outlier cars |

**Changes to Columns:**

| **Column** | **Change Applied** |
| --- | --- |
| make | One-Hot Encoded |
| modelyear | Scaled (0 to 1) |
| transmission | One-Hot Encoded |
| Body Type | One-Hot Encoded |
| Fuel Type | One-Hot Encoded |
| km | Scaled (0 to 1) + Outliers removed |
| ownerno | Scaled (0 to 1) |
| prize | Scaled (0 to 1) |
| city | One-Hot Encoded |

**Handling Missing Values:**

Missing values are common in real-world datasets and can be handled in different ways depending on the data type.

* **Numerical Columns**: For columns like **'modelyear'**, **'km'**, **'ownerno'**, and **'Prize'**, we used **median imputation**. The median is used because it is less sensitive to outliers compared to the mean. This is important for columns like **'km'** and **'Prize'**, where extreme values can significantly affect the mean.
* **Categorical Columns**: For the **'Body Type'** column, which is categorical, we used **mode imputation**. The mode is the most frequent value in the column, making it a reasonable choice when handling missing categorical data.

**Standardizing Data Formats:**

* The **'km'** column was checked to ensure it was in a consistent format. The values were converted to integers, and any non-integer or string-based representations (e.g., '50,000 kms') were stripped of the unit ('kms') and then converted into numerical values. This step ensures that the data is uniformly structured for analysis and modeling.

**Encoding Categorical Variables:**

Machine learning models generally require numerical data, so we must convert categorical data (e.g., **'make'**, **'transmission'**, **'Body Type'**, **'Fuel Type'**, **'city'**) into numerical formats.

* **One-Hot Encoding**: For nominal categorical variables (those without a natural order like **'make'**, **'transmission'**, **'Body Type'**, **'Fuel Type'**, and **'city'**), **one-hot encoding** was used. This technique creates new binary columns for each category, with a '1' indicating the presence of that category and a '0' otherwise. For example, if a car has a **'make'** of 'Audi', a new column for 'Audi' would have a '1', and all other 'make' columns would have '0'.
* The **drop\_first=True** parameter in the pd.get\_dummies() function ensures that the first category of each column is dropped to avoid multicollinearity (i.e., redundant information).

**Normalizing Numerical Features:**

Normalization is important when working with models sensitive to the scale of the data, such as distance-based models like K-Nearest Neighbors (KNN).

* **Min-Max Scaling**: We applied **Min-Max scaling** to the numerical columns **'modelyear'**, **'km'**, **'ownerno'**, and **'Prize'**. This technique scales the features to a fixed range, typically between 0 and 1. It ensures that all features contribute equally to the model, preventing features with larger ranges from dominating the learning process.

**Removing Outliers:**

Outliers can distort the training of machine learning models, especially for models that assume data points are normally distributed.

* **Interquartile Range (IQR) Method**: This technique is used to identify and remove outliers. We calculate the **IQR** for the numerical features **'km'** and **'Prize'**:
  + **IQR** is the range between the 25th percentile (Q1) and the 75th percentile (Q3) of the data.
  + Outliers are identified as values outside the range [Q1−1.5×IQR,Q3+1.5×IQR][Q1 - 1.5 \times \text{IQR}, Q3 + 1.5 \times \text{IQR}][Q1−1.5×IQR,Q3+1.5×IQR].
  + Any data points that fall outside this range are removed from the dataset, as they are considered outliers.

**Saving the Processed Dataset:**

After completing all the preprocessing steps, the final dataset is saved as a new CSV file. The processed dataset is now ready for model training.

**Summary of Key Preprocessing Steps:**

* **Data Cleaning**: Removed irrelevant columns and handled missing values using median (for numerical) and mode (for categorical) imputation.
* **Feature Engineering**: Ensured consistent data formats, particularly for numerical features like **'km'**.
* **Encoding**: Converted categorical variables into numerical ones using one-hot encoding.
* **Normalization**: Scaled numerical features to a range between 0 and 1 using Min-Max scaling.
* **Outlier Detection**: Removed outliers using the Interquartile Range (IQR) method.