For each of these selected features:

| **Feature** | **Preprocessing Step** |
| --- | --- |
| km | Remove commas/text, convert to numeric, scale |
| ownerNo | Impute missing, convert to integer |
| modelYear | Impute, convert to int, calculate car\_age |
| engineCC | Clean, impute, convert, scale |
| price | Clean, keep in lakhs (no scale), impute |
| ft | One-hot encode |
| transmission | One-hot encode |
| make | Label encode or one-hot (depending on cardinality) |
| model | Label encode or group if too many unique values |
| bt | One-hot encode (SUV, Sedan, etc.) |
| car\_age | Derived from modelYear, scale |

* Clean values
* Handle missing data
* Encode categorical features
* Normalize where required

| **Step** | **Columns** | **Technique** | **Purpose** |
| --- | --- | --- | --- |
| Keep price as-is | price(in lakhs) | No scaling | It's the target value |
| Fill missing values | km, engineCC, etc. | Median (numeric), Mode (cat) | Prevent null-related errors |
| Normalize features | km, ownerNo, engineCC, etc. | Min-Max Scaling | Puts numeric features in range [0, 1] |
| Encode categories | ft, bt, model, City, etc. | One-Hot Encoding | Convert text to machine-friendly format |
| Merge final set | All cities | concat() | Ready for training & EDA |

All city-wise datasets are **fully preprocessed**, including:

* Cleaning (km, engineCC, price)
* Imputation (median fill for numeric fields)
* Encoding:
  + One-hot for fuel type, transmission, body type
  + Label encoding for make and model (if available)
* Normalization (for numeric features like km, ownerNo, etc.)
* Derived field: car\_age
* city column added