Data Preprocessing

1. Read Dataset

- Load the dataset (e.g., using pandas.read_csv()).
- Inspect the data structure (.head(), .info(), .describe()).

---

2. Explore the Data

1. **Basic Exploration**:
    - Print the first few rows to understand the structure of the dataset.
    - Drop irrelevant columns like IDs if they don't contribute to the target variable.
2. **Check Datatypes**:
    - Identify numerical and categorical columns.
    - Convert categorical columns to the category datatype for better memory efficiency.
3. **Categorical Column Analysis**:
    - Display the number of unique categories in each categorical column.
4. **Missing Values**:
    - Check for missing values in each column.
    - Calculate the percentage of missing values.

---

3. Handle Missing Values

1. **High Null Ratios**:
    - Drop columns where the null-value percentage is too high (e.g., >50%).
2. **Categorical Columns**:
    - Fill missing values with the mode of the column.
3. **Numerical Columns**:
    - Visualize the distribution of each column (e.g., using histograms or skewness statistics).
    - If skewed, fill missing values with the **median** to reduce the effect of outliers.
    - For symmetric distributions, use the **mean** for imputation.
4. **Validate Null Handling**:
    - Recheck the dataset to ensure no missing values remain.

---

4. Outlier Detection and Treatment

1. **Visualize Outliers**:
    - Use box plots to detect outliers in numerical columns.
2. **Capping Outliers**:
    - Replace values above the upper whisker with the maximum non-outlier value (upper bound).
    - Replace values below the lower whisker with the minimum non-outlier value (lower bound).
3. **Categorical Outliers**:
    - For rare categories (low frequency), replace them with the mode of the column.

---

5. Check for Duplicates

- Remove duplicate rows using drop_duplicates().

---

6. Drop Low-Variance Columns

- Remove columns with very low variance (e.g., standard deviation close to zero).

---

7. Feature and Label Separation

- Split the dataset into:
    - **Features (X)**: All independent variables.
    - **Label (y)**: Target variable.

---

8. Encoding Categorical Columns

1. **Ordinal Data**:
    - Use label encoding.
2. **High-Cardinality Columns**:
    - Use binary encoding or frequency encoding.
3. **Low to Medium Cardinality Columns (3-6 categories)**:
    - Use one-hot encoding to represent these categories.

Model Building

1. Split Data

- Divide the dataset into training and testing subsets (e.g., 80% train, 20% test).
- Use train_test_split() from sklearn.

---

2. Train the Model

- Select a model based on the problem:
  - **Classification**: Logistic Regression, Decision Trees, Random Forests, etc.
  - **Regression**: Linear Regression, Decision Trees, Random Forest Regressor, etc.
- Train the model on the training dataset.

---

3. Evaluate the Model

1. **For Classification**:
   - Use:
     - **Confusion Matrix** to understand True Positives, True Negatives, False Positives, and False Negatives.
     - **F1-score** to balance precision and recall.
     - **Accuracy** to measure overall correctness.
2. **For Regression**:
   - Use:
     - **Mean Absolute Error (MAE)**: Average magnitude of errors.
     - **Mean Squared Error (MSE)**: Penalizes larger errors more than smaller ones.
3. **For Clustering**:
   - Use silhouette score.