2. Handle Missing Data

Why It's Important

Missing data can bias results or reduce model performance. Handling it effectively ensures data integrity.

Steps to Handle Missing Data

1. **Identify Missing Values**:
    - Look for null, NaN, or placeholder values like `999` or `-1`.
    - Summarize missing data as percentages or counts for each column.
    - Example: If 30% of entries in a column are missing, consider if that feature is still valuable.
2. **Decide How to Handle Missing Data**:
    - **Remove Rows/Columns**:
        - If a row has too many missing values (>50%), consider removing it.
        - If a column has >30% missing data and no easy way to impute, consider dropping it.
    - **Imputation**:
        - For numerical data:
            - Mean or median for symmetric distributions.
            - Mode for categorical or binary data.
        - For categorical data:
            - Most frequent value or "Unknown" category.
    - **Domain-Specific Handling**:
        - For example, forward-fill or backward-fill for time-series data ensures continuity.

Best Practices:

- Impute missing values only in the training set to avoid data leakage.
- Document imputation methods for reproducibility.

---

3. Remove Duplicates

Why It's Important

Duplicate data skews analysis by overrepresenting patterns. Removing duplicates ensures accurate modeling.

Steps to Handle Duplicates:

1. **Identify Duplicates**:
   - Check rows where all columns have identical values.
   - For partial duplicates, focus on key columns like IDs or combinations (e.g., customer ID and timestamp).
2. **Decide on Removal**:
   - Retain only the first occurrence of a duplicate.
   - If duplicates have slight differences (e.g., timestamps), use domain knowledge to decide which one to keep.
3. **Verify After Removal**:
   - Ensure important records were not mistakenly dropped.
   - Keep a backup before removing duplicates for safety.

Best Practices:

- Check for duplicates after merging datasets or concatenating rows.
- Log removal criteria for transparency.

---

4. Handle Outliers

Why It's Important

Outliers can distort model predictions, especially in algorithms sensitive to scale, like linear regression.

Steps to Handle Outliers:

1. **Detect Outliers**:
   - Statistical methods:
     - **Interquartile Range (IQR)**:
       - Define outliers as values < Q1 - 1.5*IQR or > Q3 + 1.5*IQR.
     - **Z-Score**:
       - Treat values with |z| > 3 as outliers.
   - Visual methods:
     - Box plots, scatter plots, or histograms.
2. **Decide on Outlier Handling**:
   - **Remove**:

        - For data entry errors or when the outlier is irrelevant to the analysis.
    - **Transform**:
        - Apply logarithmic, square-root, or similar transformations to reduce the effect.
    - **Cap Values**:
        - Set thresholds to replace extreme outliers with fixed values (e.g., 1st and 99th percentiles).
    - **Analyze Separately**:
        - If outliers are valuable (e.g., high-spending customers), analyze them in a separate context.

Best Practices:

- Treat each feature independently for outlier detection.
- Use domain expertise to determine if an outlier is truly erroneous or meaningful.

---

5. Fix Data Types

Why It's Important

Incorrect data types cause errors during analysis or modeling. Fixing them ensures consistency and compatibility.

Steps to Fix Data Types:

1. **Inspect Data Types**:
    - Check all columns for mismatched types (e.g., dates stored as strings or numerical IDs stored as text).
2. **Convert Data Types**:
    - **Numerical Conversion**:
        - Convert numerical strings (e.g., "123.45") to float or integer.
    - **Date Conversion**:
        - Transform date strings (e.g., "2024-12-28") into datetime objects for easier analysis.
    - **Categorical Conversion**:
        - Ensure categorical features are stored as appropriate data types for better memory usage and compatibility with encoding.
3. **Verify After Conversion**:
    - Validate the changes by summarizing the dataset to check for errors.

    - Perform a quick analysis (e.g., histograms or unique values) to confirm correctness.

Best Practices:

- Consistently format similar features (e.g., prices in dollars and cents).
- Use metadata or domain expertise to ensure proper type assignments.